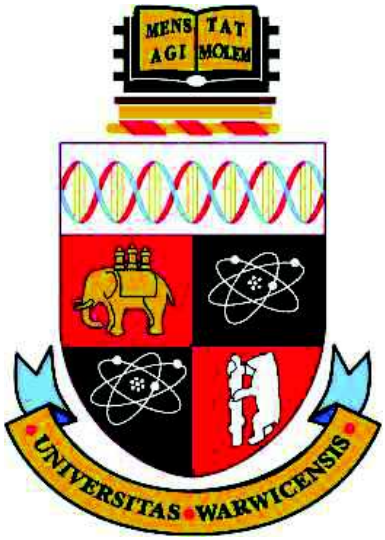


The Dynamic Chain Event Graph

by

Rodrigo Abrunhosa Collazo

A thesis submitted for the degree of

Doctor of Philosophy in Statistics

University of Warwick, Department of Statistics

April 2017

THE UNIVERSITY OF WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

My thanks go first to professor Jim Q. Smith who has been an encouraging, friendly and zealous supervisor. His stimulating and insightful advice has inspired me to bring this work to fruition. His constant availability, patience and openness for discussions have played a key role in this thesis and represent his generosity and his work ethnic as regards his students and colleagues.

I wish to thank the 15-month and 24-month panel members, Dr. David Rossell, Prof. Jane Hutton and Prof. Simon French for their important feedback. In particular, Dr. David Rossell provided valuable comments on non-local priors. I am also grateful to John Horwood and the CHDS research group for providing the CHDS data set and to Prof. James Henry and Dr. Geraldine Mcleod for allowing me to present the results associated with the process of booking a tourist train. Thank you, too, to all the members of the Statistics Department, especially the Administrative staff, for constructing an astonishing and inspiring research environment.

I am very privileged to have had many important and encouraging teachers throughout my academic life, especially in the Brazilian Naval Academy and in the Federal University of Rio de Janeiro (UFRJ/COPPE). Special thanks to Professors Antonio Luiz Porto e Albuquerque, Basílio de Bragança Pereira, Hélio dos Santos Migon and Mozart Menezes for their support in my pursuit of this doctorate.

I am also fortunate to have some friends to help and motivate me. In particular I would like to thank Braz Lamarca, Leonardo Antonio Monteiro Pessôa and Pier Giovanni Taranti for their forbearance during my PhD course.

This thesis would not be possible without the financial endorsement from the

Brazilian Navy and CNPq-Brazil. I directly rely on the institutional support from the Naval Secretariat of Science, Technology and Innovations (SecCTM) and the Naval System Analysis Center (CASNAV). I am especially thankful to Vice Admiral Bernardo José Pierantoni Gambôa, Rear Admiral Cid Augusto Claro Junior, Captain Marco Eugênio Madeira Di Beneditto, Captain Lucia Artusi, Captain Ana Cláudia de Paula and Dr. Ivana Cardial de Miranda Pereira for believing in my work.

Finally and most of all I thank my beloved family. To my father Angelo, *in memoriam*, and my mother Etelvina for their unconditional self-sacrifice in preparing me for life and their continual encouragement to pursue my dreams. To my loyal esteemed brother Fabio for his steady support and motivation in all moments of my life.

Declarations

I hereby declare that this work is based on my own research under the supervision of professor Jim Smith, except when stated otherwise. This thesis has not been submitted for examination at any another university. Some of this work has been published as follows: the material in Chapter 5 has been published in the *Bayesian Analysis* under the title "A new family of Non-Local Priors for Chain Event Graph model selection" (Collazo and Smith, 2016). It is also available as CRiSM Working Paper 15-02. That paper was written with Jim Q. Smith but the text in Chapter 5 is entirely my own work.

I am one of the co-authors in a preliminary paper concerning the dynamic version of Chain Event Graphs. This paper has been published in the *Eletronic Journal of Statitics* under the title "The Dynamic Chain Event Graph" (Barclay et al., 2015). It resulted from a joint work with Lorna M. Barclay, Jim Q. Smith, Peter A. Twaites and Ann E. Nicholson. It is also available as CRiSM Working Paper 14-04. A seminal DCEG paper based on Chapters 6 and 7 of this thesis is currently being revised for submission. This work is co-authored with Jim Q. Smith.

Part of the material in Chapters 3 and 4 will be published in a book about CEG models co-authored with Jim Q. Smith and Christiane Görgen. A paper associated with the booking process of a tourist train is presently under preparation. This applied work is based on the material described in Sections 3.1, 3.2 and 4.5 and is co-authored by Geraldine Mcleod, James Henry, Jordan van der Klei, Jim Q. Smith and John Horwood. I performed all computational experiments for this thesis using my own R package for Chain Event Graph models. This constitutes a joint work

with Pier G. Taranti. A public version of this package and its corresponding paper are currently under revision for release and publication, respectively.

Abstract

The Chain Event Graph (CEG) is a type of tree-based graphical model that accommodates all discrete Bayesian Networks as a particular subclass. It has already been successfully used to capture context-specific conditional independence structures of highly asymmetric processes in a way easily appreciated by domain experts.

Being built from a tree, a CEG has a huge number of free parameters that makes the class extremely expressive but also very large. Exploring the enormous CEG model space then makes it necessary to design bespoke algorithms for this purpose. All Bayesian algorithms for CEG model selection in the literature are based on the Dirichlet characterisation of a family of CEGs spanned by a single event tree. Here I generalise this framework for a CEG model space spanned by a collection of different event trees. A new concept called hyper-stage is also introduced and provides us with a framework to design more efficient algorithms.

These improvements are nevertheless insufficient to scale up the model search for more challenging applications. In other contexts, recent analyses of Bayes Factor model selection using conjugate priors have suggested that the use of such prior settings tends to choose models that are not sufficiently parsimonious. To sidestep this phenomenon, non-local priors (NLPs) have been successfully developed. These priors enable the fast identification of the simpler model when it really does drive the data generation process. In this thesis, I define three new families of NLPs designed to be applied specifically to discrete processes defined through trees. In doing this, I develop a framework for a CEG model search which appears to be both robust and computationally efficient.

Finally, I define a Dynamic Chain Event Graph (DCEG). I develop object-recursive methods to fully analyse a particularly useful and feasibly implementable new subclass of these models called the N Time-Slice DCEG (NT-DCEG). By exploiting its close links with the Dynamic Bayesian Network I show how the NT-DCEG can be used to depict various structural and Granger causal hypotheses about a studied process. I also show how to construct from the topology of this graph intrinsic random variables which exhibit context-specific independences that can then be checked by domain experts. Throughout the thesis my methods are illustrated using examples of multivariate processes describing inmate radicalisation in a prison, and survey data concerning childhood hospitalisation and booking a tourist train.

Abbreviations

2T-DBN	Two Time-Slice Dynamic Bayesian Network
2T-DCEG	Two Time-Slice Dynamic Chain Event Graph Network
AHC	Agglomerative Hierarchical Clustering
BF	Bayes Factor
BN	Bayesian Network
CEG	Chain Event Graph
CHDS	Christchurch Health and Development Study
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DCEG	Dynamic Chain Event Graph
fp-NLP	Full Product Non-Local Prior
LP	Local Prior
lpBF	Logarithm of Posterior Bayes Factor
MAP	Maximum A Posteriori
NLP	Non-Local Prior
NT-DBN	N Time-Slice Dynamic Bayesian Network
NT-DCEG	N Time-Slice Dynamic Chain Event Graph
NT-SDCEG	N Time-Slice Stratified Dynamic Chain Event Graph
OAHC	Optimise Agglomerative Hierarchical Clustering
pBF	Posterior Bayes Factor
PC	Point of Contact
pp-NLP	Pairwise Product Non-Local Prior
pm-NLP	Pairwise Moment Non-Local Prior

pMOM-NLP	Product Moment Non-Local Prior
SCEG	Stratified Chain Event Graph
SS	Sample Size

Chapter 1

Introduction

1.1 Motivation

Graphical models are useful tools that facilitate the interaction among scientists of different areas, and between these and decision makers. They provide a visual framework focusing on the structural relations that characterize processes. This interface is particularly attractive since its essence can be appreciated even by laypeople with little mathematical training. The Bayesian network (BN) (Pearl, 1988, Neapolitan, 2004, Cowell et al., 2007, Korb and Nicholson, 2011, Smith, 2010) is the most widely used type of graphical model in the statistical domain. It has been applied in diverse areas such as: management, business, environmental studies, military applications, computational engineering, biology, medicine, genetics, pedigree analyses and many others.

However despite its flexibility and power to describe a wide range of processes, a BN also has some well-documented limitations (Poole and Zhang, 2003). For example, such problems are always described using a preassigned collection of random vectors living on a product space and have to hold for all levels of the conditioning random vectors. In many domains it has been discovered that this yields a rather restrictive family of hypotheses. In practice we often find that distinct levels of variables can give rise to distinct collections of relevant variables showing distinct types of dependences. In particular, a BN model cannot fully de-

pict context-specific conditional independences (Spiegelhalter and Lauritzen, 1990, Boutilier et al., 1996), i.e. where conditional independences hold only for certain values of the conditioning probability vector.

To build classes of models that can accommodate such assumptions, various non-graphical methods have now been suggested and appended to the BN framework, including context-specific BNs (Boutilier et al., 1996, Poole and Zhang, 2003, McAllester et al., 2008) and object-oriented BNs (Koller and Pfeffer, 1997, Bangsø and Wuillemin, 2000). A graphical limitation of BNs is illustrated in Example 1. This describes a very simple version of the process associated with the refugee crisis in the Mediterranean Sea.

Example 1 (Refugee Crisis). *In the European refugee crisis, thousands of migrants lost their lives in the Mediterranean Sea trying to travel from North Africa to Southern European using flimsy boats. Suppose that we would like to model the chance (variable C) of a migrant arriving alive in Europe (y- alive, n- dead) as a function of the roughness of the sea (variable S). The variable S measures the mean wave height of the one third highest waves according to five categories: a- less than 0.1m; b- between 0.1 and 1.25m; c- between 1.25m and 4m; d- between 4m and 6m; and e- greater than 6m.*

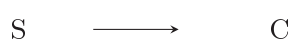


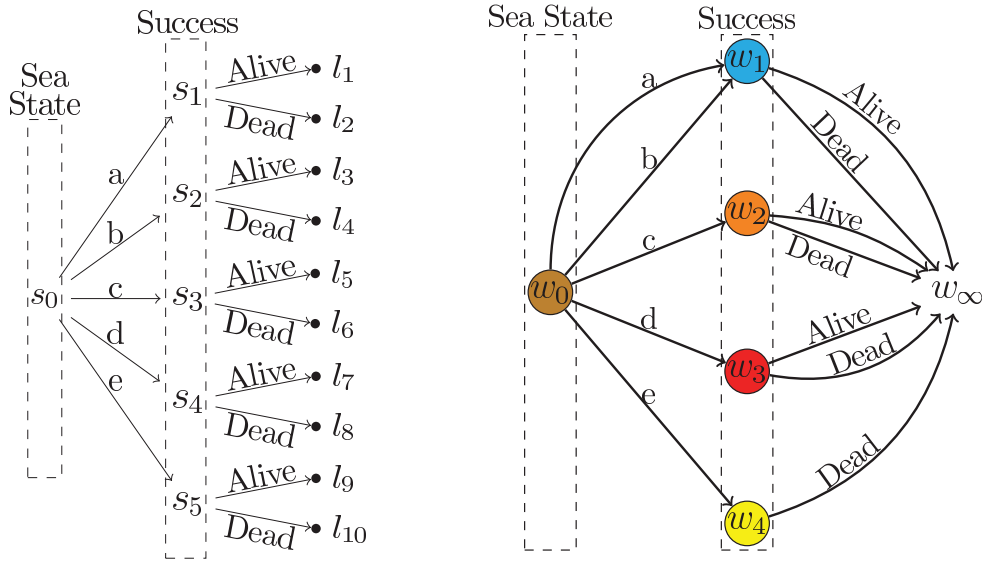
Figure 1.1: The BN model corresponding to the chance (variable C) of a migrant arriving alive in Europe as a function of the roughness of the Mediterranean sea (variable S).

Assume that the probability of success decreases monotonically as the wave height increases and that there is no difference between categories a and b. Then Figure 1.1 shows the BN that represents this process. Suppose its hypothetical conditional probability table is given by: $P(C = y|S = a) = P(C = y|S = b) = 0.15$*;* $P(C = y|S = c) = 0.05$*;* $P(C = y|S = d) = 0.01$*;* $P(C = y|S = e) = 0$*. Observe that without adding dummy random variables we are not able to express graphically the context-specific statements associated with categories a and b. Also note that it is not possible to read directly from the graph that the probability of*

success is indeed zero for category e. □

An alternative class of graphs which is able to at least depict structural asymmetries directly is an event tree; see Shafer (1996). By embellishing this structure with colours based on a probabilistic measure we obtain a probabilistic tree that provides the basis of a graphical framework called a Chain Event Graph (CEG) (Smith and Anderson, 2008, Thwaites et al., 2008, Smith, 2010). For further discussion, see Example 1 (cont.) below.

Example 1 (Refugee Crisis - cont.). The event tree in Figure 1.2a depicts the multiple ways that the migration process can unfold for each refugee. Note that at this point domain experts do not need to consider variables or the conditional independence relationships between them. The tree supports a modelling representation that allows them to focus instead on different qualitative descriptions of the process.



(a) Event Tree

(b) CEG

Figure 1.2: The Event Tree and CEG model corresponding to the chance of a migrant arriving alive in Europe as a function of the roughness of the Mediterranean sea.

Figure 1.2b depicts the corresponding CEG. Note that even without a deep mathematical understanding of the semantics of a CEG, we can intuitively read from

the CEG that the conditional probability of success is identical given sea state a or b. The probability zero of success associated with a sea state category e is directed incorporated into the CEG model by omitting its corresponding edge. These graphical properties play an important role in more complex processes and, particularly, in many practical dynamic scenarios where the probability conditional tables tend to contain more structural zeros. □

The class of CEG models is closely related to the probabilistic decision graphs (Bozga and Maler, 1999, Jaeger, 2004, Jaeger et al., 2006) and encompasses the entire discrete BN class (Smith and Anderson, 2008). It also provides a useful framework for learning under complete sampling (Freeman and Smith, 2011a) model selection (Freeman and Smith, 2011a, Barclay et al., 2013) and causal analyses (Thwaites et al., 2010, Cowell and Smith, 2013). In addition to this it also supports efficient propagation of new information (Thwaites et al., 2008, Thwaites and Smith, 2006b) and studies with missing data (Barclay et al., 2014).

A probabilistic tree typically has a huge number of free parameters. This richness of models makes the class of CEGs extremely expressive, particularly when the underlying tree is asymmetric (French and Insua, 2010). However the space of models is consequently immense even for problems described by a fairly moderate number of random variables. In order to find useful CEG models it is therefore necessary to design bespoke model search algorithms. This can require us to make use of various heuristic methods (Silander and Leong, 2013, Cowell and Smith, 2014). Dirichlet conjugate learning for CEGs (Freeman and Smith, 2011a) provides us with a natural and efficient framework to do this because the score functions of different models can be expressed analytically and in closed form.

However it has been discovered fairly recently that although standard Bayes factor search automatically penalises complex models, the extent of this penalisation is not great enough to preclude the choice of a too large model, even when supporting data is rich (Dawid, 1999, 2011, Johnson and Rossell, 2010). Various methods have therefore been developed to address this issue. Because of the enormous size

of our model space, any method which biases the model search to simpler models can turn out to be extremely useful.

Another contentious issue common with all Bayesian search methods is that the values of the hyper-parameters of standard Dirichlet priors need to be set to initialise the model search. Although, at least from a subjectivist perspective, this is not a problem in terms of Bayesian estimation, it is a problem when addressing model selection, since it would be impossible – common with all Bayesian search methods – in practice to reflect on the massive number of appropriate values of possible explanatory model hyperparameter vectors individually.

One common way to attempt to circumvent these issues is to adopt a vague prior. However, it has been known for some time that this reduces the robustness of the model selection by making the result dependent on the parameter that sets the vagueness of beliefs a priori (Rao and Wu, 2001, Berger and Pericchi, 2001, Pericchi, 2005). This and related instabilities have also been reported for a graphical model in the BN context (Steck, 2008, Silander et al., 2007, Steck and Jaakkola, 2002). Such instabilities can also occur when searching the class of CEG models if care is not taken especially when using standard conjugate local priors BF model selection (Silander and Leong, 2013, Collazo and Smith, 2016). In this thesis, both to obtain parsimonious models and to guarantee the stability of the model selection I will develop new families of prior distributions specifically designed for discrete processes supported by trees.

In many real-world settings it has also became increasingly evident that describing a process directly through components of a multivariate time series enables us to obtain more accurate and well-calibrated models. There has been previous interest in developing dynamic classes of CEGs. For example, Freeman and Smith (2011b) proposed a CEG multi-process model where different cohorts of units entering the system at different discrete time points are accompanied during only one time interval. Although trees are identical across all the cohorts the transition probabilities are allowed to shift dynamically. In contrast here, a dynamic CEG

is designed for a different purpose: to describe how a single cohort of units who arrive in the system simultaneously might evolve over successive time points under the hypothesis of time-homogeneity after some time T.

Of course, the Dynamic Bayesian Network (DBN) provides a well-established graphical framework for discrete processes that develop over time: see Dabrowski and de Villiers (2015), Rubio et al. (2014), Marini et al. (2015), Li et al. (2014), Sun and Sun (2015), Khakzad (2015). A DBN corresponds to an extension of a BN for modelling and reasoning within dynamic systems whose progress is recorded over a time sequence. However despite their flexibility, because they are based on a Directed Acyclic Graph (DAG) a DBN has the same drawbacks that a BN has.

So, it is clearly time to develop a Dynamic Chain Event Graph (DCEG) to model longitudinal data. Tree-based graphical models provide a powerful graphical framework through which sequences of events that happen over time can be directly accommodated. For example, each possible time sequence can be described by a particular path in the event tree. This enables domain experts to express their beliefs about dynamic process in terms of events rather than random variables. It also provides graphical support for embodying logical constraints and context-specific statements that may change over time and managing sparse conditional probability tables without requiring additional dummy or degenerate variables.

In Barclay et al. (2015) a very general dynamic class of CEG models was defined. Despite this generality the methods in that paper did not provide a formal framework for systematically *constructing* a DCEG and *reading* the conditional independences it entailed. Furthermore, these authors admitted that model selection over this immense class of models was extremely challenging. This arose because the DCEG model space could be huge even for quite small problems. To develop bespoke algorithms to search this massive model space for explanatory and causal mechanisms in moderately sized problems therefore requires us first to define useful and pertinent subclasses of DCEGs to search over.

From the above, in this thesis I aim at addressing the following research questions:

1. Can we construct a family of prior distributions that ensure parsimony and stability of CEG model selection to the setting of hyper-parameters, particularly when greedy model search algorithms are used?
2. How can we formally define a general class of DCEGs in discrete time? How can we systematically construct it?
3. Is there a useful subclass of DCEGs? How can we interpret it? What are their properties?

Below I will outline the thesis plan for approaching these research questions.

1.2 Thesis Outline

In Chapter 2, I will briefly review the concepts that support the construction of graphical models. I next focus on discrete BN models since this model class is one of the most well-established graphical frameworks in the scientific community. I will present their graphical semantics and the Bayesian methods for learning an appropriate graph from complete data using a characterisation of Dirichlet priors. Finally, I will introduce the DBNs and highlight some important limitations of BNs and DBNs through specific examples.

Of course, there are several methods for learning graphical models and making inferences within them. Perhaps the two most principled frameworks adopt either a relative frequency approach or a Bayesian approach to probability, see e.g. the discussion in Neapolitan (2004) and Cowell et al. (2007). Throughout this thesis I have chosen the latter for three main reasons. First, I am inclined to use the Bayesian methodology for various technical reasons outlined in Howson and Urbach (1994), Krause and Clark (1994), Lindley (1994) and O'Hagan and Forster (2004).

Second, in the context within which I am working, data is observational and there may be some domain information which needs to be added to the processes. A Bayesian framework then enables us to accommodate prior domain knowledge straightforwardly through an appropriate choice of prior. Third, at least when

running the models at certain settings within the examples used in this thesis I find that many cells are empty or sparse. Again through using a graphical model this gives rise to no technical issues other than the fact that the prior drives the inference and comparisons can be performed homogeneously through the model space.

Chapter 3 will be dedicated to the CEG framework. I will explain how to construct a CEG model and how to use it to explore the conditional independences that may be present in the data. I will then discuss the conjugate Bayesian learning based on Dirichlet priors (Freeman and Smith, 2011a) and the advantages of propagating evidence using a CEG model. At this point I will propose a modified and more efficient version of the propagation algorithm developed by Thwaites and Smith (2006b). All these concepts will be further illustrated using a new real-world process of booking a tourist train and an extended version of the example on kidney and liver disorders first analysed by Thwaites and Smith (2006b).

In Chapter 4 I will discuss the CEG model search algorithms developed in the literature. I will first review Bayes Factor model selection and a formal Dirichlet characterisation of CEG models when the graphs in the CEG model space share the same event tree. I will then propose an extension of this framework for a CEG space model which is spanned by a collection of event trees. In order to do this I assume two further conditions that are often adopted for BN model selection. I will also review a useful family of CEG models called Stratified CEGs (SCEGs) (Cowell and Smith, 2014). The SCEG models constitute an important CEG class because it contains all discrete BNs and context-specific BNs (Boutilier et al., 1996, Poole and Zhang, 2003, McAllester et al., 2008) as a special case. The SCEG subclass enables us to explore many plausible collections of explanatory hypotheses even though it is smaller than the full CEG class.

I will then proceed to explore two strategies to find the maximum a posteriori (MAP) CEG model, i.e. the CEG model which a posteriori appears to be the most probable explanation of the data generating process. I will first present a

greedy model search algorithm developed by Freeman and Smith (2011a) and analyse some possible ways of improving it. I will then propose a modified version of this algorithm that is able to search the CEG model space more efficiently. Through this development a new concept called *hyper-stages* is appended to the CEG framework. As I will show, a hyper-stage structure enables us not only to design a more efficient algorithm but also to embellish the qualitative description of our models by accommodating domain hypotheses within the model search.

I will then review a dynamic programming algorithm (Bellman, 1957) for SCEG models as presented in Cowell and Smith (2014). Although this method guarantees that the MAP CEG will be found, those authors recognised that it can easily become unable to explore CEG model spaces of fairly moderate sizes and that the development of reliable approximative algorithms is needed. I will discuss these challenges and propose some strategies for minimising them. Finally, I will revisit the booking train example and explore the space of possible explanatory hypotheses for this process using the dynamic programming model search algorithm. The results have already been reported in Collazo et al. (2016).

The next three chapters constitute the most original and relevant methodological contributions of this thesis. In Chapter 5, I will review the model selection methods based on non-local priors (NLPs) (Johnson and Rossell, 2010, Consonni et al., 2013, Consonni and La Rocca, 2011, Johnson and Rossell, 2012, Altomare et al., 2013, Rossell and Telesca, 2015). Embodying a separation measure between nested models within their constructions, these priors automatically penalise complex models although these are still consistent in the Bayesian framework. For this reason, an NLP is more able to retrieve a parsimonious model than standard priors such as those based on conjugate learning when the simpler model is truly the source of the data generation process.

I will then propose three new families of NLPs for CEG model selection: the full product NLPs (fp-NLPs), the pairwise product NLPs (pp-NLPs) and the pairwise moment NLPs (pm-NLPs). I will examine some undesirable phenomena that may

occur when standard Dirchlet priors and product NLPs are used in conjuction with some greedy model search algorithm. I will argue that pm-NLPs provide a promising and simple way to render the model search more robust and to identify parsimonious CEG models in high-dimensional settings where conditional dependence structures tend to be sparse. I will then proceed to develop a CEG model search framework that will allow us to use pm-NLPs in conjunction with my modified greedy model search algorithm introduced in the previous Chapter.

To explore empirically the good properties of pm-NLPs, I will conduct extensive computational experiments for CEG model searches using two real-world examples and an R package for CEGs that I have collaboratively developed (Collazo and Taranti, 2016). First I will revisit a well-studied data set on childhood hospitalisation (Fergusson et al., 1981, 1984, 1986, Barclay et al., 2013, Cowell and Smith, 2014). My method will be shown to provide us with more robust results to the setting of hyper-parameters than standard Dirchlet priors. It will also be able to give new insights into the dynamic between the risk of a child being hospitalised and the socio-economic factors that characterise his family background.

To illustrate how these selection methods scale up to large problems I will look at the process of radicalisation of inmates in British prisons. This topic has become a high priority for policymakers and an issue of lively public debate since experts have recognised the fundamental role that prisons can have as a hot house for radicalisation. Indeed, some empirical evidence indicates that extremist ideologists can be highly compelling amongst certain classes of inmates.

To model the radicalisation process we have to address a number of technical challenges and domain particularities. First, the radicalisation dynamic is often characterised by many context-specific conditional independences and asymmetric developments. Second, the classes of each relevant variable involved are remarkably unbalanced. The percentage of radical prisoners - those individuals of special interest - is also very tiny. Third, the collection of possible models is extremely large. To find a useful model it is therefore necessary to design bespoke effi-

cient algorithms that combine prior domain information and data. In this thesis, the explanatory variables can capture the ethnological and criminal background of prisoners as well as their prison networks. For all these reasons this application was actually beyond the scope of the search algorithms discussed in Chapter 4. Note that a summary of Chapter 5 has already been published in Collazo and Smith (2016).

In Chapters 6 and 7, I will define a new discrete multivariate dynamic model, the *Dynamic Chain Event Graph* (DCEG), that is a natural counterpart of a Chain Event Graph. I contributed to the first paper that introduced these processes (Barclay et al., 2015). To advance the foundation of this new model class in discrete time here I have focused on the following three objectives:

1. To develop a practical methodology to guide the construction of effective DCEGs in practice using new classes of DCEGs which can coherently distribute domain judgements as symmetries across different teams of experts.
2. To introduce new graphical semantics to express context-specific conditional independence structures often only implicit in the composed domain beliefs elicited during the knowledge engineering process.
3. To develop formal methods to identify the random processes intrinsic to an elicited probabilistic structure, and also to explore the implicit conditional independence structures between these processes.

Throughout these Chapters all concepts will be illustrated and further discussed using an example that models dynamically a simple radicalisation process of a prison population. This development has already been reported in Collazo and Smith (2016) and submitted for review.

In Chapter 6 I will define a very general class of DCEG models that is able to handle discrete longitudinal data observed at regular time intervals. For this purpose, in Section 6.1 I will develop a new object approach to elicit a process using an infinite tree. The objects will be defined according to the temporal structure and domain information associated with the corresponding process. This approach will enable

graphical modellers to incorporate time-invariant variables in a DCEG model and to split the modelling task between different domain expert teams. This ensures the consistency of the composite model. In Section 6.2, I will introduce a formal framework to embed a probabilistic map on an infinite tree and discuss some topological concepts on probabilistic trees. In the concluding Section 6.3 I will formally define a DCEG and give a general representation of a finite DCEG in terms of a particular graphical periodicity and time-homogeneity.

In Chapter 7, I will introduce a new model class I have devised called N Time-Slice Dynamic Chain Event Graph (NT-DCEG). Such a DCEG class will have the potential for modelling several different processes in real-world applications. It will also enable us to develop bespoke algorithms to search the massive DCEG model space for useful explanatory models and causal mechanisms.

A formal link between a Markov state-transition diagram and an NT-DCEG will also be constructed as well as some links between DCEGs and DBNs. In particular I will prove that a Two Time-Slice Dynamic Bayesian Network (2T-DBN) can always be expressed as a 2T-DCEG. I will then explain how a particular set of CEGs can be used to derive a DCEG model. The connection between a DCEG and a corresponding set of CEGs will be then analysed.

I will also show how implicit conditional independence relationships encoded in an NT-DCEG can be read from its representation using the graphical concept of a cut. This will enable us to identify from the topology of an NT-DCEG convenient sets of random variables - often not immediately apparent from the original description - whose relationship captures critical conditional independences embedded within the described process. Smith and Anderson (2008) have argued that the cuts in a CEG can be used as an alternative framework to answer queries corresponding to the Pearl's d-separation theorem (Pearl, 1988) in a BN. This idea can be naturally extended to an NT-DCEG. Finally I will explore the ideas of local, contemporaneous and stochastic independences. These have a strong link with notions of Granger noncausality (Granger, 1969). See also Hsiao (1982),

Geweke (1984), Eichler (2007) and Eichler and Didelez (2010).

In the concluding chapter I will summarise the contributions of this thesis and discuss further research streams that may stem from it.

Chapter 2

Graphical Models

I start this chapter reviewing graphical models and graph theory. I then proceed to discuss various statistical properties corresponding to one of the most popular graphical models, the Bayesian Network (BN), and its dynamic counterpart, the Dynamic BN. I finish by outlining some of the limitations of BN models.

2.1 An overview of Graphical Models

In order to use graphical models, it is necessary first to define the semantics. This connects the graphical symbols with the theoretical or applied domain. In other words, the semantic lexis establishes the function, scope and use of each symbol. Therefore, the potential and limits of any graphical model are given by this initial semantic definition.

Graphical models become statistical models when their semantics relates the topology of the graph to the probability measure associated with a class of probability models; a condition which is satisfied for all graphs considered here. I here restrict myself to discrete model graphs. That is, I only consider problems whose observed random variables are discrete, although the parameters defining these variables can be (and usually are) continuous quantities.

I also restrict my focus to graphical models whose topologies include a graph $\mathscr{G} = (V, E)$, where V and E are, respectively, the set of vertices and edges. The

set V depicts the main elements of a domain as stated by the qualitative modelling; for example, variables, situations, decisions and so on. The set E represents the relationship among the previous elements according to the semantic lexis. In general, this set introduces the probabilistic map over the model. The topology may have other elements such as colours and different shapes of vertices, but the graph provides the framework which is then populated with for these additional elements.

In a broad sense, the use of a graphical model in a specific domain relies on the following steps: domain modelling; learning of the model; and inference and manipulation (Cowell et al., 2007). The first step corresponds to the qualitative modelling and requires us to answer the following questions: i) Which objects in the domain constitute the vertices?; and ii) How do these elements relate to each other, i.e. which edges constitute the set E? This early step typically demands a deep knowledge of the field and, consequently, the active participation of the domain experts. Within this phase, the pictorial representation of the model helps the interaction between the experts and the statistical specialists, implicitly embedding the problem within the statistical representation.

In the second phase, the statistician needs to populate the model with probabilistic distributions. This quantitative elicitation can be obtained by adopting a subjective or an objective-based modelling approach. In the first case the analyst uses some formal framework to map the experts' prior beliefs into subjective probabilities (Wright and Ayton, 1994) associated with the random variables defined in the qualitative model. Here, the graphical model is an important and valuable tool to help experts elicit their knowledge a priori. It helps to prevent experts from losing focus in a welter of tangential technicalities or freezing when they address high levels of complexity. However, those probability distributions can also be defined using exclusively a data set. This corresponds to adopt an objective approach.

Of course, both approaches, the subjective and objective ones, can be partially combined. In complex real-world problems this mixed strategy is often adopted

because it is rather difficult to obtain sufficient reliable data in order to conduct the data-driven learning process for the whole model. Another reason is that embedding domain information allows the analyst to obtain models that are more appealing for decision makers and domain experts. In many settings this approach also helps to reduce the modelling complexities and to keep the computational costs under control.

Remember that data can drive not only the definition of probabilistic distributions in the second phase but also the specification of the set of edges that constitute the qualitative structure of the model: answer the second question in the first phase. The intrinsic properties of the graph are used to carry out model selection and model learning, optimizing the performance of the chosen model or pool of models. In a strict objective approach the analyst is almost the only actor responsible for the learning process and the interaction with the domain expert rarely happens.

Finally, the statisticians will analyse the obtained results and present the initial feedback to the domain experts. Here, the graph is a helpful resource to stimulate the experts' interaction and to underpin the subsequent rounds of analysis, inference and manipulation. It enables the domain experts to consider the results in a very intuitive and direct way, thus helping the statistician to translate very advanced technical concepts into an objective and common language.

2.2 Introduction to Graph Theory

In this section I review some concepts of graph theory that will be important for the development of this thesis. For further details, see e.g. Netto (2006), Diestel (2006) and Cowell et al. (2007).

Definition 1 (Graph)**.** A *graph* $\mathbb{G}$ is a pair $\mathbb{G} = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ is a set of vertices and E is a set of edges $(v_i, v_j) \in V \times V$ between the vertices in V. If V and E are both finite sets the graph is said to be *finite*. Otherwise it is said to be *infinite*.

Definition 2 (Subgraph)**.** A graph $\mathbb{G}_s = (V_s, E_s)$ is a *subgraph* of $\mathbb{G} = (V, E)$ if $V_s \subseteq V$ and $E_s \subseteq E$.

Definition 3 (Undirected and Directed Graphs)**.** An *undirected graph* is a graph $\tilde{\mathbb{G}} = (\tilde{V}, \tilde{E})$ whose edges have no directionality and are called *undirected edges*. In this case each undirected edge $(v_i, v_j) \in \tilde{E}$ is depicted as a line between v_i and v_j. In contrast, a *directed graph* $\mathbb{G} = (V, E)$ is a graph whose every edge (v_i, v_j) has an orientation. A *directed edge* $(v_i, v_j) \in E$ is graphically represented as a arrow from v_i to v_j. We can obtain an undirected version $\tilde{\mathbb{G}} = (\tilde{V}, \tilde{E})$ of a directed graph $\mathbb{G} = (V, E)$ by removing the directionality of every edge in $\mathbb{G}$. Explicitly, we then have $\tilde{V} = V$ and $\tilde{E} = \{(v_i, v_j), (v_j, v_i); (v_i, v_j)$ or $(v_j, v_i) \in E\}$.

Definition 4 (Labelled Graph)**.** A *labelled graph* $\mathbb{G}$ is a graph $\mathbb{G} = (V, E, R_V)$, where $R_V = \{(v, l_v); v \in V\}$ is the set of labels l_v over each vertex in V, $E = \{(v_i, v_j, l_e); (v_i, v_j) \in V \times V\}$ is the set of labelled edges such that the triad (v_i, v_j, l_e) represents an edge (v_i, v_j) with label l_e.

Definition 5 (Multi-Graph)**.** A graph $\mathbb{G} = (V, E)$ is said to be a *multi-graph* if there can be two or more edges (v_i, v_j) between any two vertices v_i and v_j in V, otherwise the graph is called *simple*.

Definition 6 (Parent and Child)**.** In a directed graph $\mathbb{G} = (V, E)$ a vertex $v \in V$ is a *parent* (or a *child*) of a vertex $v_n \in V$ if $(v, v_n) \in E$ (or $(v_n, v) \in E$). Let $pa(v_n) = \{v \in V; (v, v_n) \in E\}$ denote the parent set of a vertex $v_n \in V$ and $ch(v_n) = \{v \in V; (v_n, v) \in E\}$ denote the children of a vertex $v_n \in V$.

Definition 7 (Adjacent Vertices)**.** Two vertices v_i and v_j of a graph $\mathbb{G}$ are said to be *adjacent* if there is at least one edge between them.

Definition 8 (Walk, Path and Trail)**.** In a graph $\mathbb{G} = (V, E)$ a *walk* of length L is a sequence of vertices $(v_{i_0}, \ldots, v_{i_L})$ such that every edge $(v_{i_k}, v_{i_{k+1}})$, $k=0, \ldots, L-1$, pertains to E. A walk that no two vertices are repeat is called a *path*. A *trail* of length L is a sequence of distinct vertices $(v_{i_0}, \ldots, v_{i_L})$ such that at least one of the edges $(v_{i_k}, v_{i_{k+1}})$ and $(v_{i_{k+1}}, v_{i_k})$ pertains to E, for all $k = 0, \ldots, L-1$. In a

trail $(v_{i_0}, \ldots, v_{i_L})$ of a directed graph $\mathbb{G} = (V, E)$ a vertex $v_k, k = 0, \ldots, L-1$ is called a *collider vertex* if the edges $(v_{i_{k-1}}, v_{i_k})$ and $(v_{i_{k+1}}, v_{i_k})$ pertain to E.

Definition 9 (Descendent and Ancestor vertices)**.** Take a DAG $\mathbb{D} = (V, E)$. A vertex v is a *descendent* of a vertex v_a in $\mathbb{D}$ if there exists a v_a-to-v path but there is not a v-to-v_a path. Conversely, v is an *ancestor* of a vertex v_a in $\mathbb{D}$ if there exists a v-to-v_a path but there is not a v_a-to-v path. A subset $V_* \subseteq V$ is called an *ancestral set* if for every $v \in V_*$ we have that $pa(v) \subseteq V_*$. Let $An(V_*)$ be the smallest ancestral set that contains V_*.

Definition 10 (Connected Graph)**.** A graph $\mathbb{G} = (V, E)$ is said to be *connected* if there is a trail between every pair of its vertices. A *component* of a graph $\mathbb{G}$ is a maximal connected subgraph of $\mathbb{G}$.

Definition 11 (Cycle)**.** In a graph $\mathbb{G} = (V, E)$ a *cycle* of length L is a sequence of vertices $(v_{i_0}, \ldots, v_{i_L})$ such that $(v_{i_0}, \ldots, v_{i_L-1})$ is a path and such that $v_{i_0} = v_{i_L}$ and $(v_{i_{L-1}}, v_{i_L}) \in E$.

Definition 12 (Directed Acyclic Graph)**.** A *directed acyclic graph* (DAG) is a directed graph without cycles. A DAG $\mathbb{D} = (V, E)$ yields an ordering

$$O(V) = (v_{i_1}, \ldots, v_{i_N})$$

over the vertex set $V = \{v_1, \ldots, v_N\}$ such that for any two vertices v_{i_a} and $v_{i_b}, b > a$, there does not exist an edge (v_{i_b}, v_{i_a}) in E. If for every vertex v_{i_n}, $n = 1, \ldots, N-1$, the edge (v_{i_n}, v_{i_j}), $j = n+1, \ldots, N$, is in E, then the DAG is said to be complete.

Definition 13 (Tree)**.** A *tree* $\mathcal{T} = (V, E)$ is a connected simple graph whose undirected version has no cycles. Any vertex in V can be designated as the root vertex v_0 and the tree is said to be *rooted*. A *leaf vertex* is a non-root vertex of a tree that has only one adjacent vertex. In a rooted tree a vertex v is said to be at a *level* ℓ_d if the v_0-to-v path has length d. In this thesis all trees are assumed to be directed and rooted. This implies that the root vertex v_0 has no parents whilst the rest of the vertices in the tree have only one parent. In this case, a leaf vertex l has no children.

Definition 14 (Star)**.** A tree that has at maximum one vertex connected to two or more other vertices is called a *star*.

Definition 15 (Forest)**.** A *forest* is a graph whose every component is a tree.

2.3 Introduction to Bayesian Networks

A BN model is a pair $\mathbb{B} = (\mathbb{D}, \mathscr{P})$, where $\mathbb{D} = (V, E)$ is a DAG and $\mathscr{P}$ is a probabilistic measure. Recall that in a DAG $\mathbb{D} = (V, E)$, the set of vertices is given by a well-ordered set $V = \{v_1, \ldots, v_n\}$, where each vertex v_i represents a variable $\mathcal{Z}_i$, and the edge set E contains a collection of directed edge $(v_i, v_j), i < j$. The edges in E enable analysts to describe whether a particular variable provides any relevant probabilistic statement to explain another variable given a set of contextual information.

In a BN, the encoding of probabilistic hypotheses is possible due to the concept of conditional independence. For example, take three discrete random variables Z_1, Z_2 and Z_3 whose set of categories are, respectively, $\mathbb{Z}_1 = \{1, \ldots, L_1\}$, $\mathbb{Z}_2 = \{1, \ldots, L_2\}$ and $\mathbb{Z}_3 = \{1, \ldots, L_3\}$. If a domain analyst believes that collecting information on the value of Z_2 once the value of Z_3 is known brings no further improvement to explain variable Z_1, then Z_2 is not probabilistically relevant to variable Z_1 given the value of Z_3. In this case we say that Z_1 is conditionally independent from Z_2 given Z_3. This implies that for every triad (z_1, z_2, z_3) in $\mathbb{Z}_1 \times \mathbb{Z}_2 \times \mathbb{Z}_3$, we have that $p(Z_1 = z_1 | Z_2 = z_2, Z_3 = z_3) = p(Z_1 = z_1 | Z_3 = z_3)$. This idea directly generalises for random vectors and continuous probabilistic measures as formally stated in Definition 16.

Definition 16 (Conditional Independence)**.** Take three random vectors $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in a probability space $(\Omega, \mathscr{A}, \mathscr{P})$. We say that $\boldsymbol{X}$ is *conditionally independent* of $\boldsymbol{Y}$ given $\boldsymbol{Z}$ under $\mathscr{P}$, and write $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} | \boldsymbol{Z}$, if and only if for every set $A \in \mathscr{A}$ the probability $P(\boldsymbol{X} \in A | \boldsymbol{Y}, \boldsymbol{Z})$ is measurable with respect to a function of $\boldsymbol{Z}$

alone, i.e.

$$\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z} \iff P(\boldsymbol{X} \in A|\boldsymbol{Y}, \boldsymbol{Z}) = P(\boldsymbol{X} \in A|\boldsymbol{Z}), \tag{2.1}$$

whenever $p(\boldsymbol{y}, \boldsymbol{z})$ is strictly positive.

Let $\boldsymbol{\mathcal{Z}}^{(m)} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_m\}$ denote the first m variables of a set of ordered random variables $\boldsymbol{\mathcal{Z}} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_N\}$ in a probability space $(\Omega, \mathscr{A}, \mathscr{P})$. Also let $pa(Z_j) = \{Z_i \in \boldsymbol{\mathcal{Z}}^{(j-1)}; v_k \in pa(v_j)\}$ be the parent set of Z_j with respect to $\mathbb{D}$. We can now introduce a useful Markov property that enables us to relate a probability measure to a graphical topology.

Definition 17 (Ordered Markov Property (OMP))**.** Take a set of ordered random variables $\boldsymbol{\mathcal{Z}}$ in a probability space $(\Omega, \mathscr{A}, \mathscr{P})$ and a DAG $\mathbb{D}$. The probability measure $\mathscr{P}$ satisfies the *ordered Markov property* relative to $\mathbb{D}$ if for every pair of non-adjacent vertices v_i and v_j in V, $i < j$, a variable Z_j is conditionally independent of a variable $Z_i, i < j$, given its parent set $pa(Z_j)$.

Now we can use the OMP to formally define a BN model.

Definition 18 (Bayesian Network)**.** A *Bayesian Network* (BN) is a graphical model constituted by a set of random variables $\boldsymbol{\mathcal{Z}}$ in a probability space $(\Omega, \mathscr{A}, \mathscr{P})$ and by a DAG $\mathbb{D}$ such that the probability measure $\mathscr{P}$ satisfies the ordered Markov property relative to $\mathbb{D}$.

Example 2 below presents a naive BN model to describe the radicalisation process of inmates in a prison system.

Example 2 (Radicalisation Process). *A model of a male prisoner's radicalisation within prisons uses as explanatory variables his social networks and how the population is affected by prison transfers (Hannah et al., 2008, Neumann, 2010, Silke, 2011). The physical movements and social interactions of prisoners are constantly being monitored and recorded. Here the radicalisation process is summarised by a variable R that classifies a prisoner into one of three categories: resilient to (r), vulnerable to (v) or adopting (a) radicalisation. A social network (variable N) can*

take one of the following three levels: s- sporadic, f- frequent, i- intense. These levels measure the frequency that a "standard" prisoner is able to socially interact with other prisoners who are identified as potential recruiters to radicalisation. A binary variable T records whether an inmate remain in the prison (n) or is transferred (t) to another prison.

Figure 2.1: The BN associated with Example 2

Assume that the variable Transfer T is independent of the variable Network N given the variable Radicalisation R. Consider the hypothesis that all those prisoners who have not adopted radicalisation are equally likely to be transferred. Figure 2.1 depicts a possible BN to represent this process. □

Larger collections of conditional independence structures than those described by the OMP can be read from a BN model using the following properties satisfied by the ternary conditional independence relation (Dawid, 1979, Spohn, 1980):

Symmetry $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z} \Rightarrow \boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X}|\boldsymbol{Z}$

Decomposition $\boldsymbol{X} \perp\!\!\!\perp (\boldsymbol{Y}, \boldsymbol{W})|\boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z}$ and $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W}|\boldsymbol{Z}$

Weak Union $\boldsymbol{X} \perp\!\!\!\perp (\boldsymbol{Y}, \boldsymbol{W})|\boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|(\boldsymbol{Z}, \boldsymbol{W})$

Contraction $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z}$ and $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W}|(\boldsymbol{Y}, \boldsymbol{Z}) \Rightarrow \boldsymbol{X} \perp\!\!\!\perp (\boldsymbol{Y}, \boldsymbol{W})|\boldsymbol{Z}$

These four properties constitute the semi-graphoid axioms. These allow analysts to explore the relevance of information using a graphical topology initially elicited (Pearl and Paz, 1987). If the probability measure $\mathscr{P}$ is strictly positive then the fifth properties given below also holds and we have a graphoid. For an intuitive interpretation of the graphoid axioms see Pearl (2009).

Intersection $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z}, \boldsymbol{W}$ and $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W}|(\boldsymbol{Y}, \boldsymbol{Z}) \Rightarrow \boldsymbol{X} \perp\!\!\!\perp (\boldsymbol{Y}, \boldsymbol{W})|\boldsymbol{Z}$

An alternative way to read conditional independences is to use the d-separation theorem initially stated in Pearl (1986, 1988) and then more formally treated in

Verma and Pearl (1990) and Geiger and Pearl (1990). To review this result, take two vertices v_a and v_b of a DAG $\mathbb{D} = (V, E)$ and any subset $V_S \subset V \backslash \{v_a, v_b\}$. A trail τ between v_a and v_b is said to be *blocked* by V_S in $\mathbb{D}$ if there is a vertex $v \in \tau$ such that one of the conditions holds:

1. v pertains to V_S and v is a non-collider vertex with respect to τ; or
2. v is a collider vertex in τ but v and all its descendants are not in V_S.

If every trail between v_a and v_b is blocked than v_a and v_b are said to be d-*separated* by V_S. It then follows that two disjoint subsets V_A and V_B are said to be d-separated by a subset $V_S \subset V \backslash (V_A \cup V_B)$ if and only if every pair of vertices (v_a, v_b), such that $v_a \in V_a$ and $v_b \in V_B$, are d-separated by V_S.

Theorem 1 (d-Separation Theorem,Pearl (1986, 1988))**.** *Assume a BN model $\mathbb{B} = (\mathbb{D}, \mathscr{P})$, where $\mathbb{D} = (V, E)$ and take any three disjoint subsets V_A, V_B and V_S of V. Let $\boldsymbol{\mathcal{Z}}_A$, $\boldsymbol{\mathcal{Z}}_B$ and $\boldsymbol{\mathcal{Z}}_S$ be the set of random variables corresponding, respectively, to V_A, V_B and V_S. It then follows that*

$$\boldsymbol{\mathcal{Z}}_A \perp\!\!\!\perp \boldsymbol{\mathcal{Z}}_B | \boldsymbol{\mathcal{Z}}_S \iff V_S \text{ d-separates } V_A \text{ and } V_B. \tag{2.2}$$

The property in Equation 2.2 is often called a global Markov property.

In Lauritzen et al. (1990) and Cowell et al. (2007), the d-separation theorem is rewritten using an undirected graph $\mathbb{D}_M(A \cup B \cup S) = (V_M, E_M)$ corresponding to a transformation of DAG $\mathbb{D} = (V, E)$ spanned by V_A, V_B and V_S by the following steps:

1. Take the graph $\mathbb{D}_{anc} = (V_{anc}, E_{anc})$, where $V_{anc} = An(V_A \cup V_B \cup V_S)$ in $\mathbb{D}$ and $E_{anc} = \{(v_i, v_j) \in E; v_i, v_j \in V\}$.

2. Construct the graph $\mathbb{D}_M(A \cup B \cup S) = (V_M, E_M)$ from $\mathbb{D}_{anc}$, where $V_M = V_{anc}$ and $E_M = \tilde{E}_{anc} \cup E_{mar}$. $\tilde{E}_{anc}$ is the set of undirected edges corresponding to E_{anc}, i.e, $\tilde{E}_{anc} = \{(v_i, v_j); (v_i, v_j) \in E_{anc}\}$. E_{mar} is the set of undirected edges between any pair of vertices (v_i, v_j), $i < j$, in V_M, such that in $\mathbb{D}_{anc}$ the vertices v_i and v_j have at least one common child vertex and $v_j \notin ch(v_i)$.

In an undirected graph $\mathbb{G} = (V, E)$ V_S is said to separate V_A and V_B, where $V_A \cup V_B \cup V_S$ are any three disjoints subsets of V, if every path between any pair of vertices (v_a, v_b), $v_a \in V_a$ and $v_b \in V_B$, passes through V_S. The criterion of d-separation as presented in Theorem 1 can then be restated as follows:

$$\boldsymbol{\mathcal{Z}}_A \perp\!\!\!\perp \boldsymbol{\mathcal{Z}}_B | \boldsymbol{\mathcal{Z}}_S \iff V_S \text{ separates } V_A \text{ and } V_B \text{ in } \mathbb{D}_M(A \cup B \cup S). \tag{2.3}$$

This alternative formulation is often more useful and appealing operationally.

Using the d-separation property domain experts can identify local conditional independence structures that potentially characterise their processes. Therefore the qualitative aspects of a process can more deeply analysed and detailed and the probability distributions embedded within a model can be more precisely elicited and calibrated. The d-separation theorem also provides us with a solid criterion with which to manipulate and factorise complex graphical structures into local graphical components with simpler topologies. These local subgraphs constitute a key aspect that enables us to design and justify efficient inference and model selection algorithms: see e.g. Cowell et al. (2007), Korb and Nicholson (2011), Neapolitan (2004) and Smith (2010).

It is also shown that in a BN model $\mathbb{B} = (\mathbb{D}, \mathscr{P})$ the probability measure $\mathscr{P}$ over the set of random variables $\boldsymbol{\mathcal{Z}}$ recursively factorizes as follows:

$$p(\boldsymbol{Z} = \boldsymbol{z} | \mathbb{D}) = \prod_{Z_i \in \boldsymbol{\mathcal{Z}}} p(Z_i = z_i | \boldsymbol{Z}_{pa(Z_i)} = \boldsymbol{z}_{pa(Z_i)}), \tag{2.4}$$

where $\boldsymbol{Z} = (Z_1, \ldots, Z_N)$ and $\boldsymbol{Z}_{pa(Z_i)} = (Z_{i_1}, \ldots, Z_{i_k})$ are random vectors whose every component is, respectively, a random variable in $\boldsymbol{\mathcal{Z}}$ and $pa(\mathcal{Z}_i)$. This also implies that in a BN model $\mathbb{B} = (\mathbb{D}, \mathscr{P})$ every variable is conditionally independent of its non-descendent variables with respect to $\mathbb{D}$ given its parent set. For further details see e.g. Cowell et al. (2007).

2.4 Introduction to Dynamic Bayesian Networks

In its most common formulation (Dean and Kanazawa, 1989, Kjærulff, 1992, Nicholson, 1992) a Dynamic Bayesian Network (DBN) models the temporal rela-

tionship among variables that are observed at regular time intervals. So henceforth in this thesis we let $\boldsymbol{\mathcal{Z}}(t)$ be a set of random variables $\boldsymbol{\mathcal{Z}}$ observed at time-interval t and whose total ordering is not necessarily identical over time.

Assume that a DAG $\mathbb{D}(T)=(V(T),E(T)))$ represents the conditional independence relationships between the components of $\boldsymbol{\mathcal{Z}}(T)$. Now define the set of temporal edges $E_{\dagger}(T)$. These are edges from a vertex $v_i(t) \in V(t), t < T$, to a vertex $v_j(T) \in V(T)$ and so represent relationships between variables in different time-slices. Note that there might be a temporal edge $(v_i(t), v_i(T))$. This would depict the dependence of a variable $\mathcal{Z}_i$ at time T on its value at any previous time t, $t < T$. Inherinting the usual semantics of a BN, two non-adjacent vertices $v_i(t) \in V(t)$ and $v_j(T) \in V(T)$, such that $t \leq T$ and, if $t = T$, $i < j$, then imply that $\mathcal{Z}_j(T)$ is conditionally independent of a variable $\mathcal{Z}_i(t)$ given its parent set $pa(Z_j(T))$, where $pa(Z_j(T)) \subseteq \cup_{k=0}^{T-1}\boldsymbol{\mathcal{Z}}(k) \cup \boldsymbol{\mathcal{Z}}^{(j-1)}(T)$.

Therefore, a DBN model for the first T time-intervals consists of a probability measure $\mathscr{P}$ associated with the set of random variables $\cup_{t=0}^{T}\boldsymbol{\mathcal{Z}}(t)$ and a DAG $\bar{\mathbb{D}}(T)=(\bar{V}(T),\bar{E}(T))$, where $\bar{V}(T)=\cup_{t=0}^{T}V(t)$ and $\bar{E}(T)=\cup_{t=0}^{T}(E(t)\cup E_{\dagger}(t))$. Without further assumptions, the specification of a DBN model is challenging since for each time-slice t a different DAG $\mathbb{D}(t)$ and its corresponding temporal edge set needs to be defined. So for practical reasons two additional conditions are often hypothesised. The first of these is to assume a Markov condition of order $N-1$. This demands that the values of a variable at time t depend only on the values of variables at the last $N-1$ previous and current intervals. The second common hypothesis is to assume that the process is time-homogeneous.

These assumptions greatly simplifies the specification of the models. This is because we only have to elicit N conditional probabilities tables, one for each of the first $N-1$ time-slices and another for the succeeding time-slices. Therefore, to obtain a DBN we only need to define a limited number of DAGs and temporal edges: the DAGs $\mathbb{D}(t), t = 0,\ldots,N-2$, for the first $N-1$ time-slices and their corresponding sets of temporal edges $E_{\dagger}(t)$; and a DAG $\mathbb{D}(t) \equiv \mathbb{D}, t = N-1, N, \ldots$, for

all subsequent intervals and its corresponding set of temporal edges $E_{\dagger}(t) \equiv E_{\dagger}$. When these two additional assumptions are adopted a DBN is called a N Time-Slice DBN (NT-DBN). A common choice in practice is to set $N = 2$, see e.g. Korb and Nicholson (2011), Neapolitan (2004), Pourret et al. (2008). This implies that the current value of a given variable may persist in the system at maximum one time-slice ahead. In this case, the state of the system at time $t + 1$ is completely determined by the values of its variables at time t. This simplification provides satisfactory result particularly in systems that evolve slowly over time and if we are interested in filtering and forecasting over short-term time horizon.

Example 3 (Dynamic Radicalisation Process). *Radicalisation has a psychosocial dynamic, which is naturally modelled as a process developing over time (Hannah et al., 2008, Neumann, 2010, Silke, 2011, Christmann, 2012, Guittet et al., 2012, Schmid, 2013, Demetriou et al., 2014). Here suppose that the counts of this process are recorded weekly and that the hypotheses in Example 2 are still valid for each time-slice. Being an isolated environment, a change in its underlying mechanisms only happens rarely in the short to medium term: the very recent events are the main psychosocial drivers of the prison population. In this scenario, it is plausible that the time-homogeneous and 1-Markov conditions might hold. So, to define the temporal edges assume that all variables at time $t + 1$ depend only on their previous value and the value of the variable Transfer T at time t.*

A prisoner with an extreme political or religious ideology typically constructs social networks that are unlikely to moderate him and might well reinforce his current beliefs. In this case, his social contacts reflect his personal vision of the world and his disengagement from militant extremism often requires personal incentives only available after he leaves the prison. On the other hand, the conversion of a non-radical prisoner to an extremist ideology can well be driven by his social contacts within the prison regardless of whether he is resilient or vulnerable. Under this reasoning the social network might act on the prisoner's belief system but not the other way around.

A 2T-DBN corresponding to this description is depicted in Figure 2.2. Note

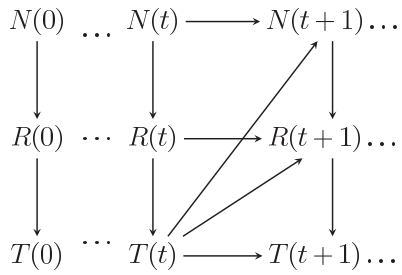


Figure 2.2: 2T-DBN associated with Example 3

that the type of psychological models described above are often called escalation models (Wiktorowicz, 2004, Moghaddam, 2005, Silber and Bhatt, 2007, Gill, 2007, Precht, 2007, Audit Commission, 2008, McCauley and Moskalenko, 2008). □

2.5 Learning the parameters of Bayesian Networks

In some contexts decision makers and domain experts can fully elicited the graphical structure of a BN model and a graphical modeller needs only to use the data to learning it. However in many settings even the topology of the BN model needs to be learnt. For this purpose in a Bayesian framework a natural choice is to compare the posterior probability of each candidate BN model $\mathbb{B}$. For example, take a BN model over a set of random variables $\boldsymbol{\mathcal{Z}} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_N\}$, where each variable Z_i, $i = 1, \ldots, N$ can assume L_i values. Let $\boldsymbol{Z} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_I)$ be a random vector associated with I different units in the system, where $\boldsymbol{Z}_i = (\mathcal{Z}_{i1}, \ldots, \mathcal{Z}_{iN})$ is a random vector that collects the values of each variable in $\boldsymbol{\mathcal{Z}}$ for a unit i, $i = 1, \ldots, I$. From the Bayes' rule the posterior probability of a model $\mathbb{B}$ is then given by

$$p(\mathbb{B}|\boldsymbol{z}) \propto p(\boldsymbol{z}|\mathbb{B})p(\mathbb{B}). \tag{2.5}$$

Thus an analyst has to specify probability distributions for his prior belief that the BN $\mathbb{B}$, $p(\mathbb{B})$, is the true model and for the marginal likelihood of $\mathbb{B}$, $p(\boldsymbol{z}|\mathbb{B})$. Recall that the model space of all possible BN models grows exponentially with the total number of variables in $\boldsymbol{Z}$. Therefore the construction of such probability

distributions for each model in the BN model space can easily get intractable without further assumptions. In fact BN learning has been proved to be a NP-Hard problem in its general formulation (Cooper, 1990, Shimony, 1994).

Fortunately BN model selection is feasible by adopting some mild assumptions over both distributions $p(\mathbb{B})$ and $p(\boldsymbol{z}|\mathbb{B})$. Here I will review how to learn a single discrete BN and its dynamic counterpart using a Bayesian approach. This is based on the characterisation of Dirichelet distributions (Heckerman et al., 1995), see also Heckerman and Geiger (1995), Geiger and Heckerman (1997) and Heckerman (1999, 2008). The elicitation of prior distributions over the model space and further details about Bayesian model selection will be reviewed in Chapter 3 when I introduce CEG model selection methods.

Let $\mathbb{Q}_n = \{q_{ni}; i = 1, \ldots, Q_n\}$ denote the set of possible configurations q_{ni} of values that the parents of a variable $Z_n \in \boldsymbol{\mathcal{Z}}$ can take, where $Q_n = \prod_{Z_j \in pa(Z_n)} L_j$. From equation 2.4 we can see that it is necessary to specify a conditional probability distribution for each q_{ni}, $n = 1, \ldots, N$,$i = 1, \ldots, Q_n$. Remember that in a Bayesian framework this can be done by putting a probability measure on each conditional probability q_{ni} itself. Thus let the set of random vectors $\boldsymbol{\Pi}_n = \{\boldsymbol{\pi}_{n1}, \ldots, \boldsymbol{\pi}_{nQ_n}\}$, $n = 1, \ldots, N$, denote a collection of random vectors $\boldsymbol{\pi}_{ni} = (\pi_{ni1}, \ldots, \pi_{niL_n})$, $i = 1, \ldots, Q_n$, such that π_{nij}, $j = 1, \ldots, L_n$, is the probability that a variable $Z_n \in \boldsymbol{\mathcal{Z}}$ takes value k given that its parent set has configuration q_{ni}.

Now consider the following conditions:

Global Independence Random vectors associated with different variables are mutually independent.

Local Independence Random vectors associated with the same variable are mutually independent.

Parameter Modularity If a variable $Z_n \in \boldsymbol{\mathcal{Z}}$ has the same parent set in two different BN models then the set of random vectors $\boldsymbol{\Pi}_n$ is the same for both models.

Complete Random Sampling The observations $\boldsymbol{Z}_i$, $i = 1, \ldots, I$, can be ex-

pressed as independent and identically distributed.

Structural Possibility For any given variable order a BN model corresponding to the complete DAG has strictly positive probability, i.e. $p(\mathbb{B}) > 0$.

Likelihood (or Markov) Equivalence The prior distributions over the parameter spaces of any two BN models that represent exactly the same set of conditional independence statements are identical.

Assuming the conditions above, Heckerman et al. (1995) showed that the prior and posterior distributions of each parameter $\boldsymbol{\pi}_{ni} \in \boldsymbol{\Pi}_n$, $n = 1, \ldots, N$ and $i = 1, \ldots, Q_n$, are inevitably Dirichlet distributions with hyper-parameters $\boldsymbol{\alpha}_{ni}$ and $\boldsymbol{\alpha}_{ni} + \boldsymbol{x}_{ni}$, respectively. Here $\boldsymbol{x}_{ni} = (x_{ni1}, \ldots, x_{niL_n})$, where x_{nij}, $j = 1, \ldots, L_n$, denotes the number of times that the variable Z_n assumes value j in a sample $\boldsymbol{z}$ under the configuration q_{ni} of its parent set. It then follows that the marginal likelihood of a BN model is given by:

$$p(\boldsymbol{z}|\mathbb{B}) = \prod_{n=1}^{N} \prod_{i=1}^{Q_n} \frac{\Gamma(\bar{\alpha}_{ni})}{\Gamma(\bar{\alpha}_{ni} + \bar{\alpha}_{ni})} \prod_{j=1}^{L_n} \frac{\Gamma(\alpha_{nij} + x_{nij})}{\Gamma(\alpha_{nij})}, \tag{2.6}$$

where $\Gamma(\cdot)$ is the gamma function, $\bar{\alpha}_{ni} = \sum_{j=1}^{L_n} \alpha_{nij}$ and $\bar{x}_{ni} = \sum_{j=1}^{L_n} x_{nij}$. Henceforward, for any n-dimensional vector $\boldsymbol{\gamma}_i = (\gamma_{i1}, \ldots, \gamma_{in})$ we convention $\bar{\gamma}_i = \sum_{j=1}^{n} \gamma_{ij}$.

A hyper-parameter $\boldsymbol{\alpha}_{ni}$, $n = 1, \ldots, N$, $i = 1, \ldots, Q_n$, represents our prior belief about each local structure, i.e, the conditional probability of a variable $Z_n \in \boldsymbol{\mathcal{Z}}_n$ given a state q_{ni} of its parent set. Recall that the prior expectation of a parameter $\boldsymbol{\pi}_{ni}$ is given by

$$E_{\boldsymbol{\pi}_{ni}}[\pi_{nij}|\boldsymbol{\alpha}_{ni}] = \frac{\alpha_{nij}}{\bar{\alpha}_{ni}}. \tag{2.7}$$

Thus a common way to set these hyper-parameters is to elicit the expected value of $E_{\boldsymbol{\pi}_{ni}}[\boldsymbol{\pi}_{ni}|\boldsymbol{\alpha}_{ni}]$ for $\boldsymbol{\pi}_{ni}$ and then to set the strength $\bar{\alpha}_{ni}$ of our prior belief in this elicited probability vector. Of course, it is still a problem to set these hyper-parameter vectors if there is a large set of candidate BN models. In this case the parameter modularity provides us with a simple and useful framework.

Take a collection of random variables $\boldsymbol{\mathcal{Z}} = \{Z_1, \ldots, Z_N\}$. Let $\mathbb{B}_C = (\mathbb{D}_C, \mathscr{P}_C)$ be

a full BN model, where $\mathbb{D}_C = (V, E_C)$ is a complete DAG corresponding to $\boldsymbol{\mathcal{Z}}$. Set a hyper-parameter $\boldsymbol{\alpha}^C$ for this full model and assume that for all $n = 1, \ldots, N$, $i = 1, \ldots, Q_n^C$ and $j = 1, \ldots, L_n$

$$p(Z_n = j | pa(Z_n) = q_{ni}^C, \mathbb{B}_C) = \frac{\alpha_{nij}^C}{\bar{\alpha}_{nij}^C}. \tag{2.8}$$

If there is a real number α such that for all $n = 1, \ldots, N$, $i = 1, \ldots, Q_n^C$ and $j = 1, \ldots, L_n$,

$$\alpha_{nij}^C = \alpha p(Z_n = j, pa(Z_n) = q_{ni} | \mathbb{B}_C), \tag{2.9}$$

then the BN $\mathbb{B}_C$ is said to have an *equivalent sample size* equal to α. The parameter modularity then guarantees that every BN model $\mathbb{B} = (\mathbb{D}, \mathscr{P})$ over $\boldsymbol{\mathcal{Z}}$ has a hyper-parameter $\boldsymbol{\alpha}$ given by

$$\alpha_{nij} = \alpha p(Z_n = j, pa(Z_n | \mathbb{B}) = q_{ni} | \mathbb{B}_C), \tag{2.10}$$

for all $n = 1, \ldots, N$, $i = 1, \ldots, Q_n$ and $j = 1, \ldots, L_n$. Note that in equation 2.10 the parent set $pa(Z_n|\mathbb{B})$ is defined with respect to $\mathbb{B}$ although the probability measure corresponds to model $\mathbb{B}_C$.

Under the parameter modularity assumption specifying prior distributions for every BN in a model space therefore require us only to elicit the expected conditional probability tables of the full model and to define the equivalent sample size. When no prior domain information is available to guide us to set the hyper-parameters, a usual recommendation is to adopt a uniform distribution for each conditional probability in Equation 2.8 and set the equivalent sample size equal to the maximum number of categories that a variable in $\boldsymbol{\mathcal{Z}}$ can have, i.e., $\alpha = \max_{n \in \{1,\ldots,N\}} L_n$. For detailed discussions on how to set the hyper-parameter $\boldsymbol{\alpha}$, see e.g. Heckerman et al. (1995), Heckerman (1999, 2008) and Neapolitan (2004).

Learning a general DBN, where each time-slice has its own set of conditional probability tables, is completely analogous to learn a BN. However an NT-DBN requires some care in how to compute $\boldsymbol{x}$ from a complete random sample $\boldsymbol{z}$ that collect observations over $T, T \geq N$, time-slices. For this purpose, we have to calculate first $\boldsymbol{x}_*(t), t = 0, \ldots, T-1$, for each time-slice t in the same way as

a standard BN. Now note that in an NT-DBN we only have to learn N time-slices because of the Markov and time-homogeneity assumptions. So $\boldsymbol{x}$ records information of N time-slices. It then follows that $\boldsymbol{x}(t) = \boldsymbol{x}_*(t)$, $t = 0, \ldots, N-2$, and $\boldsymbol{x}(N-1) = \sum_{t=N-1}^{T} \boldsymbol{x}_*(t)$.

2.6 Limitations of Bayesian Networks

A BN model provides us with a transparent graphical framework to define a process in terms of conditional independent local structures. This facilitates the identification of relevant structural components of the process, improves the accuracy of the elicited joint probability distributions and optimises the use of computational memory and time for inferences. Despite these strengths BNs are not always the appropriate graphical model to adopt because they represent a process using a preassigned collections of random vectors. In any setting where it is artificial to describe a process directly through a set of conditional probabilities between the given components of a multivariate process then its representation using a BN can be restrictive and often difficult to justify.

A BN model is particularly not recommended when a process has at least one of the following characteristics:

1. There are some context-specific conditional independence structures (Spiegelhalter and Lauritzen, 1990, Boutilier et al., 1996), see Definition 19 below.
2. The event space is a non-product space and so a process has highly asymmetrical developments. This often happens when the state space of some variables changes or even does not exist depending on the value assumed by other variables in the probability space.

Definition 19 (Context-Specific Conditional Independence)**.** Take three random vectors $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in a probability space $(\Omega, \mathscr{A}, \mathscr{P})$. We say that $\boldsymbol{X}$ is *context-specific conditionally independent* of $\boldsymbol{Y}$ given $\boldsymbol{Z}$ under $\mathscr{P}$ if and only if for some value $\boldsymbol{z}$ of $\boldsymbol{Z}$ and for every set $A \in \mathscr{A}$ the probability $P(\boldsymbol{X} \in A | \boldsymbol{Y}, \boldsymbol{Z} = \boldsymbol{z})$ is

measurable with respect to a function of $\boldsymbol{Z} = \boldsymbol{z}$ alone, i.e.

$$\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} | \boldsymbol{Z} = \boldsymbol{z} \iff P(\boldsymbol{X} \in A | \boldsymbol{Y}, \boldsymbol{Z} = \boldsymbol{z}) = P(\boldsymbol{X} \in A | \boldsymbol{Z} = \boldsymbol{z}), \quad (2.11)$$

whenever $p(\boldsymbol{y}, \boldsymbol{z})$ is strictly positive. In this case, we write $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} | \boldsymbol{Z} = \boldsymbol{z}$.

In Example 2, without introducing new random variables the BN cannot represent graphically the context-specific hypothesis associated with the variable T given the variable R. This kind of asymmetric conditional independences can only be expressed inside the conditional probability tables of the BN or through some supplementary semantics. This fact is also true for the context-specific conditional independence statements in Example 3. Recall that the radicalisation model described here for the dynamic setting has two context-specific conditional independences in addition to that one specified in Example 2 : the variable Radicalisation R at time $t + 1$ is independent of the variable Network N at time $t + 1$ given that the variable Radicalisation R assumes the value Adopting at previous time t; and the variable R at time $t + 1$ is independent of its previous value at time t given that a prisoner did not adopt radicalisation at time t.

Also note that in the dynamic context when a prisoner is transferred the process terminates. However this kind of asymmetric development cannot be immediately read from the corresponding 2T-DBN.

Some extensions to the BN framework have been proposed to handle these issues. For example, a context-specific BN embellishes the BN/DBN models (Boutilier et al., 1996, Poole and Zhang, 2003, McAllester et al., 2008) using supplementary trees to represent the conditional probability tables that have context-specific information. In this case each variable has its own tree. Alternatively, the standard BN can be reorganised in order to depict the context-specific independences using multiple vertices associated with a single variable.

Another proposal is to use Bayesian Multinets or Similarity Networks (Geiger and Heckerman, 1996). These adopt a hypothesis variable to encode the context-specific statements over a set of random variables $\boldsymbol{\mathcal{Z}}$. For each value taken by the hypothesis variable the graphical modeller has to construct a particular BN model

called local network. The collection of these local networks constitute a Bayesian Multinet or a Similarity Network.

The main difference between those models is that in a Bayesian Multinet all local networks need to depict all variables in $\mathcal{Z}$ whilst a Similarity Network depicts only the variables that relate to the hypothesis under consideration in a particular local network. To avoid losing information for not including all variables in every local network additional local networks are required and can then be identified through a *similarity graph* that models the set of hypotheses covered by the hypothesis variable.

An advantage of a Similarity Network is that the graphical modeller is not required to quantify the conditional probabilities between variables that are not covered by the hypotheses under analysis. However in both approaches, Bayesian Multinets and Similarity Networks, a process is described by a set of networks instead of a single graph. The natural consequence is that the modelling procedure becomes more complicated and the computational complexity to encode these models increases substantially compared to a standard BN. These problems only get worse when the hypothesis variable has to represent context-specific hypotheses associated with different states of the process.

Of course context-specific BNs, Bayesian Multinets and Similarity Networks can be adapted to a dynamic setting. However the corresponding drawbacks become more pronounced and the computational complexities increase dramatically. The last issue can be minimised by embellishing a model using an abstract data types called objects that enable us to hide information by encapsulating a set of variables and related processes. The only visible part of an object from outside is its interface vertices that allow us to access the object and connect it to other objects. Being based on the object-oriented programming languages (Booch, 2007) this approach has given rise to Object Oriented Bayesian Networks (Koller and Pfeffer, 1997, Bangsø and Wuillemin, 2000). Despite being a powerful and flexible BN framework for knowledge building, the contexts-specific dependences embellished within the

model through objects tend not to be expressed graphically and so will remain hidden in conditional probability tables.

Finally another class of models that enable us to handle context-specific information is the Probabilistic Decision Graph (Bozga and Maler, 1999, Jaeger, 2004, Jaeger et al., 2006). These models were originally proposed for automated check of probabilistic expert systems and is based on ordered binary decision diagrams (Bryant, 1986). This allows a Probabilistic Decision Graph to perform efficient probabilistic inference especially in models with context-specific structures. Although there is a considerable overlap between Probabilistic Decision Graphs and BNs, the Probabilistic Decision Graph model class does not constitute a superclass of BNs. However since a Probabilistic Decision Graph has an underlying tree graph it comes close to a CEG model, which is the focus of this thesis. Smith and Anderson (2008) showed that CEG models encompass all discrete BN models and its discrete variants described above as a special subclass and are also richer than Probabilistic Decision Graphs whose semantics is actually somewhat distinct (Thwaites and Smith, 2011).

Chapter 3

A Chain Event Graph

In this Chapter I will demonstrate how the CEG framework can directly circumvent the drawbacks of BNs discussed in Chapter 2, namely modelling context-specific conditional independences and asymmetric event spaces. It will also become clear that a CEG model also retains the good properties of BNs, such as conjugate learning and efficient propagation of new information.

Setting these objectives, I will start by explaining how to construct a CEG model and how to interrogate it for conditional independences using a real-world train booking process, which is analysed here for the first time. Next I will describe a conjugate learning framework for CEGs based on Dirichlet and multinomial distributions following the developments in Freeman and Smith (2011a). I will also introduce a new improvement to the propagation algorithm of new information initially proposed by Thwaites and Smith (2006b) and Thwaites et al. (2008). CEG learning and propagation are illustrated using a simple medical example concerning liver and kidney disorders.

3.1 The Train Booking Data Set

3.1.1 Introduction

Dunedin is the oldest city in New Zealand, located at the head of Otago Harbour in the South Island. It is world-famous for its heritage buildings, cultural riches and wildlife reserves. A very popular tourist activity is to explore its scenic landscape in one-day train tour organised by the Taieri Gorge Train company. Tourists have two publicly available train options: a Pukerangi trip that can be extended until Middlemarch and a Seasider journey.

A tourist arriving by cruise ship can also opt for a train package organised by the cruise company. So cruise passengers can choose between the following two alternatives:

Option 1 Take a public train.This option enables tourists to decide between a full-day Pukerangi/Middlemarch trip or the two-hour Seasider trip. This is the most economical option for cruise passengers and it is also the most profitable one for the train company. However, if tourists choose this option, they have to go to the train station on their own or by a bus chartered by the Taieri Gorge Train company.

Option 2 Take the cruise train which is an extension of the public Pukerangi/Middlemarch train line. This option provides the most convenient experience for cruise tourists since it includes a champagne breakfast and also enables the cruise passengers to take the train directly from the wharf.

A tourist can book one of these trains in different ways. For example, he can go to a travel agency or visit an online website. He can also book directly on the cruise ships or at the Dunedin train station. For the purpose of this work, I refer to these different sites that have a physical location or an online address as Points of Contact (PCs).

The train company aims at obtaining a structural understanding about some aspects that impact the tourists' preference for a public train or a cruise train in

order to improve its marketing and commercial strategy. It focus is on passengers that visited up to five PCs before booking a train trip. In particular, the managers of Taieri Gorge Train company have two clearly stated objectives.

First, they would like to explore whether the sequence of PCs has some influence on the cruise passenger's decision for train options 1 or 2. Note that any two PC sequences can have different number of visited PCs since a tourist can decide to book a train journey in his first or in the n^{th} subsequent searched PC.

Their second objective is to assess how some selected demographic variables impact the train booking. In this study, I take into account the passenger's nationality, the passenger's age and the number of cruise trips that a passenger has already taken.

3.1.2 The Data Set

The data set has 476 tourists of which 402 arrived in Dunedin by cruise ships and the 74 others arrived by other ways such as an aircraft, a car or a bus. In my analyses, I restrict the attention to the cruise passengers for two reasons. Methodologically the small sample of non-cruise tourists does not allow us to obtain robust results because there are very few individuals associated with each category of non-cruise tourists. In terms of domain interests, this decision is also justified because the goal is to understand the train booking process associated with the cruise passengers and not all type of tourists.

During the interview a tourist could indicate up to 17 different type of PCs. However, I have decided to represent this information as a binary variable distinguishing between a tourist who goes to a PC under the control of the cruise company (s- Ship) or visits any other type of PCs (o- Others). In this case, the category Ship represents only one type of PC - a PC in the cruise ship or in the cruise company's website - and the category Others aggregates the other 16 non-Ship types of PCs.

If a model included all types of PC, it would then have fewer number of individuals

per category and a huge number of parameters to learn. These problem would get particularly worst if effects of variable interactions were modelled. So adopting a binary variable enables us to construct CEG models that are not only simpler and consequently more easily interpreted by decision makers but is also more robust.

Table 3.1 shows us that in each stage i of the PC sequence the numbers of tourists who visited a PC Ship or a PC Others are almost the same order of magnitude. The maximum number of clients visiting a PC Others at any stage i is not greater than 200. Tourists also visit the 16 different types of PCs included in the category Others in a sparse way. So disaggregating them into subcategories would cause some instability in the results since each subcategory would have only 12 tourists on average. The binary simplification is also supported by the domain particularity. This happens because the train company have different marketing strategies for PCs under the control of a cruise company and the others PCs where a potential costumer can be expected to have less restrictions to pursue their objectives.

Category	$\boldsymbol{PC_1}$	$\boldsymbol{PC_2}$	$\boldsymbol{PC_3}$	$\boldsymbol{PC_4}$	$\boldsymbol{PC_5}$	$\boldsymbol{PC_a}$	$\boldsymbol{PC_b}$	$\boldsymbol{PC_c}$
Ship	223	198	50	16	4	209	196	66
Other	179	191	85	23	1	193	193	69

Table 3.1: Number of clients that visit each Point of Contact. $PC_i, i = 1, \ldots, 5$, is the i^{th} PC visited when it is considered a sequence of five PCs. $PC_j, j = a, b, c$ is the j^{th} PC visited when it is considered only the last three visited PCs. If a client visited less than four PCs, we then have that: $PC_a = PC_1$, $PC_b = PC_2$, $PC_c = PC_3$. If a client went to four PCs, we then have that: $PC_a = PC_2$, $PC_b = PC_3$, $PC_c = PC_4$. If a client went to five PCs, we then have that: $PC_a = PC_3$, $PC_b = PC_4$, $PC_c = PC_5$.

Table 3.2 reveals that most tourists (87%) prefer booking a train when they are visiting their second or third PC. This means that although data is trustworthy the fourth and fifth stages in the PC sequence do not have sufficient individuals to support reliable results. Note that this small number of clients should also be split according their previous visited PCs. To avoid this technical problem, I therefore restricted the PC sequence to three visited PCs. This implies that

I will use only the last three PCs visited by a client who went to four or more different PCs. Observe that there is no change if a client visited less than four PCs: $PC_a = PC_1$, $PC_b = PC_2$, $PC_c = PC_3$. However, if a client went to four PCs, then PC_a, PC_b and PC_c are, respectively, the second (PC_2), third (PC_3) and fourth (PC_4) PCs that he visited. Note that in this case PC_1 is discarded. If a client went to five PCs, PC_a, PC_b and PC_c are, respectively, the third (PC_3), fourth (PC_4) and fifth (PC_5) PCs that he visited. Here we do not consider PC_1 and PC_2. An analogous transformation is also applied to variable F.

Category	F_1	F_2	F_3	F_4	F_5	F_a	F_b	F_c
Booked	13	254	96	34	5	13	254	135
Searching	389	135	39	5	0	389	135	0

Table 3.2: Number of clients that booked a train over time. $F_i, i = 1, \ldots, 5$, indicates if the client booked a train during his i^{th} visit when it is considered a sequence of five PCs. $PC_j, j = a, b, c$ indicates if the client booked a train during his j^{th} visit when it is considered only the last three PCs. If a client visited less than four PCs, we then have that: $F_a = F_1$, $F_b = F_2$, $F_c = F_3$. If a client went to four PCs, we then have that: $F_a = F_2$, $F_b = F_3$, $F_c = F_4$. If a client went to five PCs, we then have that: $F_a = F_3$, $F_b = F_4$, $F_c = F_5$.

The demographic variables are defined as follows:

1. Country (C) - a binary variable differentiating between passengers from Australia or New Zealand (l- Local), or other world regions (o - overseas);
2. Age (A) - a binary variable differentiating between young passengers (y- Young) (≤ 45), or mature passengers (m- Mature) (≥ 46); and
3. Visit (V) - a categorical variable differentiating passengers with a weak (w) (at maximum 1 cruise journey) , a moderate (m) (between 2 and 5 cruise journeys) or a strong (s) (6 or more cruise journeys) tendency to enjoy cruise ships.

Table 3.3 presents the summary statistics corresponding to these variables. Over-

all, the variables Country and Visit have a well-balanced number of individuals per category. This does not happen with the variable Age where the great majority of tourists (67%) are mature. In particular, Australian or New Zealand tourists prefer local trains (57%) whilst overseas individuals have a stronger disposition (64%) to buy a train package organised by the cruise company. The proportion (56%) of young tourists who chose public trains is almost the same to that of mature tourists who took the cruise-organised trains (55%). Passengers with low inclination for cruise journey do not apparently have any clear preference between train options 1 and 2. However, passengers with a moderate and a strong propensity for cruise journeys prefer to take a local train (58%) and a cruise-organised train (61%), respectively.

Train	**Country**		**Age**		**Visit**		
Option	Local	Overseas	Young	Mature	Weak	Moderate	Strong
1	133	62	74	121	55	81	59
2	97	110	57	150	57	58	92
Total	230	172	131	271	112	139	151

Table 3.3: Number of passengers that booked each type of train according to their nationality, age and number of prior cruise travels.

3.2 CEG Modelling and Reasoning

The representation of a process using a CEG model is obtained in three major steps: the elicitation of the event tree $\mathcal{T}$ that describes the qualitative structure supporting the process; its subsequent embellishment with colours to obtain the staged tree; and eventually its transformation into the CEG graph according to some simple graphical rules (Smith and Anderson, 2008, Thwaites et al., 2008, Smith, 2010, Freeman and Smith, 2011a). The CEG model procedure is also discussed in Barclay et al. (2013) and in Cowell and Smith (2014).

Figure 3.1 depicts an event tree corresponding to the multiple paths that a tourist can take before booking a train as described in Section 3.1. In this tree, there are three distinct types of random variables: a binary variable PC_i for the $i^{th}, i = a, b, c$, site visited; a binary variable $F_i, i = a, b, c$, (b- Booked; s- Searching) representing if a tourist booked a train at PC_i or carried on searching at other PC; and the variable train T indicating the train option.

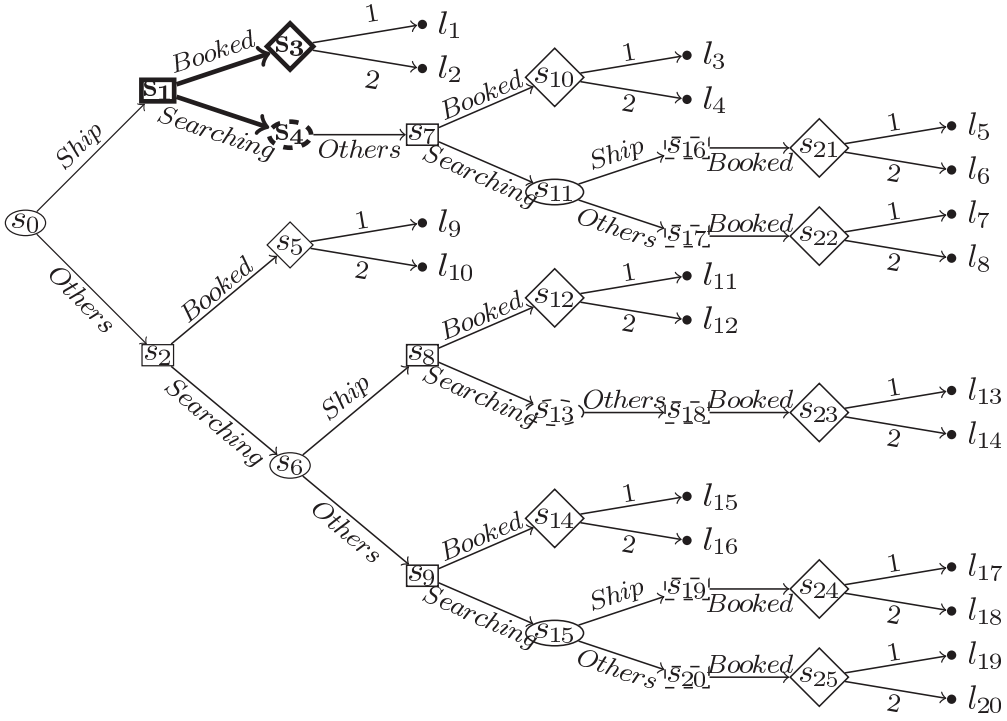


Figure 3.1: The event tree associated with a train ticket search for the last three points of contact visited by a tourism. Variables: ⬭ - PC; □ - F; ◊ - Train.

An event tree provides a flexible graphical framework that facilitates the visualisation and understanding the whole process. For these purposes, vertices associated with the same type of random variable are depicted in the event tree using an identical geometric shape. Edges associated with events with probability zero are omitted. For example, situations corresponding to variables F and PC that are given a dashed geometric shape. Since all tourists included in the sample booked a train, the variable F_3 has necessarily value equal to $Booked$.

Now observe that a tourist cannot move from a PC Ship to another PC Ship

since this movement is interpreted as a single visit to a PC Ship by the sample design. So, $PC_{i+1}, i = 1, 2$ cannot assume value $Ship$ if PC_i is equal to $Ship$. In contrast, this logical exclusion does not hold within the PC category $Others$ because this category merges 16 different types of PCs.

For instance, a tourist in the initial situation s_0 can unfold into the situation s_1 where he visits a point of contact associated with the cruiser company. He might then decide to book a train, situation s_3, or carry on searching a train ticket at an other point of contact, situation s_4.

The situation s_3 represents the state of a tourist who opts for booking a train in his visit to a PC that is linked to a cruiser company in this case. Analogously, the situation s_{25} represents an individual that books the train ticket after visiting three point of contacts that are not managed by the cruiser company.

Depicting the different ways that each unit can follow along the process under analysis, an event tree (Shafer, 1996) provides an intuitive graphical interface to translate the system dynamic into a mathematical model using plain language. Formally, the non-leaf vertices of the tree characterises a particular situation s that a unit can be at a given point during the process. It represents a transitional state from the root node to a potential end of unfolding process. Its emanating edges are associated with the events that may happen when a unit is at situation s. In contrast, a leaf vertex l, or simply a leaf l, represents a possible terminating state of the process. So, both types of vertices, a situation s and a leaf l, are defined by the consecutive events that occur along its root-to-s or root-to-l paths.

An important graphical structure corresponding to a situation s is the star subgraph of $\mathcal{T}$ called floret $\mathcal{F}(s) = (V(s), E(s))$, where the vertex set $V(s)$ is constituted by the situation s and its child situations, and the edge set includes all outgoing edges of s. For example, the floret associated with situation s_1 in Figure 3.1 is depicted in bold. Now a floret $\mathcal{F}(s)$ enables us to associate every sitution s in an event tree with a random variable $X(s)$ whose state space $\mathbb{X}(s_i) = \{e_{ij}\}$ is given by the set of events $e_{ij}, j = 1, \ldots, L_i$ that can happen once a unit is at situation s_i.

This random variable $X(s_i)$ fully represents how a process develops given that a unit is at s_i. In this way, we can implicitly obtain a probability measure yielded by the primitive conditional probabilities defined at each situation s_i as follows:

$$p(X(s_i) = e_{ij}|s) = \pi_{ij}, e_{ij} \in \mathbb{X}(s_i). \tag{3.1}$$

Embedding the set of conditional probabilities given by Equation 3.1 within an event tree enables us to obtain a probability tree. In this case, the set of situations whose state spaces are equivalent and whose conditional probabilities are identical constitute a stage. Thus, the probabilities corresponding to the emanating edges of florets rooted at any two situations that are at the same stage are hypothesised to be equal. Each stage is associated with a unique colour. When the situations in an event tree are embellished with these colours we obtain a staged tree. For formal detail about this construction, see e.g. Freeman and Smith (2011a).

Two situations s_a and s_b are said to be in the same position w if and only if they are at the same stage u and their whole subsequent unfolding processes develop under a completely analogous probability law. Thus, there is a probabilistic and graphical isomorphism between the subtrees unfolding from s_a and s_b and the subsequent evolutions of units at these situations are undistinguishable. The vertices of a CEG corresponds exactly to these positions and so the set of positions is said to be the position structure W of a CEG.

Return to the train booking example. The demographic model aims at identifying heterogeneous population of tourists according their predisposition to choose a public or cruiser-organised train using the three demographic variables: nationality, age and number of prior cruise travels. Note that in the first model the corresponding process of visiting a PC and booking a train has a very well-known and rigid sequence of possible events.

In distinction to the PC sequence model this does not happen in our second model where only a partial variable order is defined a priori. In this case, each leaf node of the demographic event tree should immediately unfolds from an event corresponding to the state space of the variable Train. This is justified because

our focus is to understand whether and how the demographic variables can explain a tourist's option for a public or cruiser-organised train. In the absence of any further domain hypotheses that enable us to elicit a complete variable order, we run a CEG model search algorithm to define it using the data available and the partial order already known.

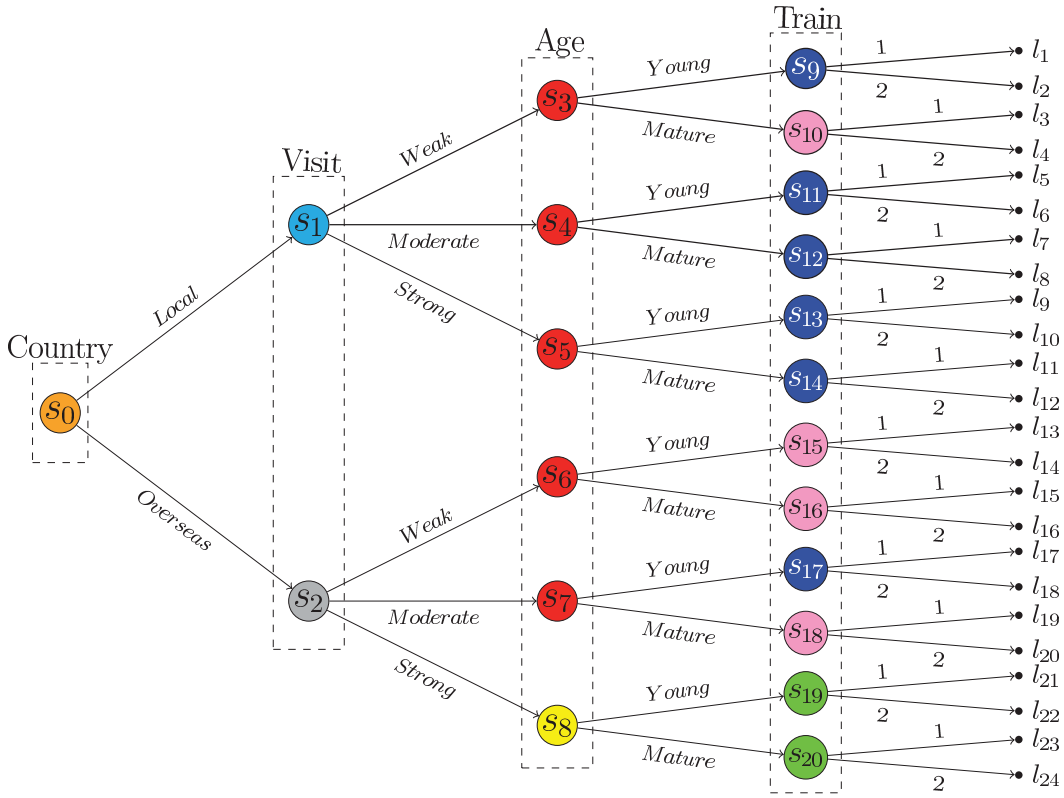


Figure 3.2: A hypothesised staged tree associated with the demographic variables and the variable Train. Variable order: $C \succ V \succ A \succ T$. The stage structure is given by: $u_0 = \{s_0\}$, $u_1 = \{s_1\}$, $u_2 = \{s_2\}$, $u_3 = \{s_3, \ldots, s_7\}$, $u_4 = \{s_8\}$, $u_5 = \{s_9, s_{11}, \ldots, s_{14}, s_{17}\}$, $u_6 = \{s_{10}, s_{15}, s_{16}, s_{18}\}$, $u_7 = \{s_{19}, s_{20}\}$.

Figure 3.2 shows a staged tree for the demographic variables associated with the train booking process under the assumed variable order $C \succ V \succ A \succ T$ and the hypothesised stage structure: $u_0 = \{s_0\}$, $u_1 = \{s_1\}$, $u_2 = \{s_2\}$, $u_3 = \{s_3, \ldots, s_7\}$, $u_4 = \{s_8\}$, $u_5 = \{s_9, s_{11}, \ldots, s_{14}, s_{17}\}$, $u_6 = \{s_{10}, s_{15}, s_{16}, s_{18}\}$, $u_7 = \{s_{19}, s_{20}\}$. Consider the situations s_5 and s_7 whose state spaces are identical,

$$\mathbb{X}(s_5) = \mathbb{X}(s_7) = \{y, m\}.$$

As these situations are coloured the same we can conclude that the probabilities of two passengers to be young given that one of them comes from local regions and has a strong tendency to take a cruise journey and the other is an overseas tourist with a moderate propensity for taking cruise trip are identical. However, these situations are not at the same position because the corresponding child situations s_{14} and s_{18} are embellished with different colours. So, the unfolding subtrees rooted at situations s_5 and s_7 are not the same since mature passengers have their train booking processes governed by quite different probability rules. On the other hand, it is easy to observe that situations s_4 and s_5 are not only in the same stage but also at the same position.

Transforming a staged tree into a CEG requires us two simple steps: to merge all situations at the same position w into a single vertex w and to divert all leaf nodes into a single sink vertex w_∞. A CEG is a compact representation of its corresponding staged tree. Therefore, it facilitates the interpretation and the readability of hypotheses embodied within the probability model by domain experts and decision makers.

The CEG model depicted in Figure 3.3 corresponds to a graphical transformation of the staged tree showed in Figure 3.2. It summarises the information depicts in the staged tree without implying any loss of information or requiring further assumptions. In doing this, the CEG graphs communicates more easily the hypothesised conditional independences structures to laypeople.

For example, the positions $w_3 = \{s_4, s_5\}$, $w_4 = \{s_3, s_7\}$ and $w_5 = \{s_6\}$ in Figure 3.3 are coloured red because the situations $s_i, i = 3, \ldots, 7$, gathered by them are all at the same stage u_3. So, the variable Age associated with a passenger is context-specific conditional independent of his corresponding variables Country and Visit given that this tourist is not at position w_6: he is not from overseas countries and does not have a strong preference for cruise trips. In contrast to BN models this kind of asymmetric conditional independences is directly and easily depicted by a CEG. Observe that all other stages in this CEG coincide with a single

position.

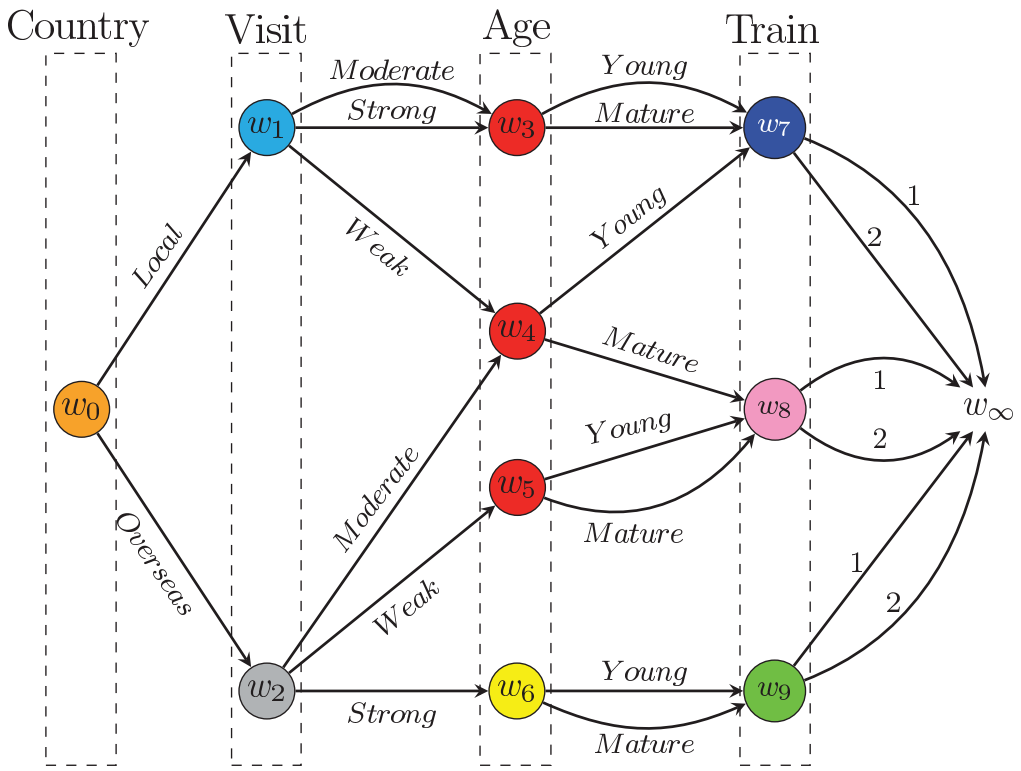


Figure 3.3: The CEG corresponding to the stage tree depicted in Figure 3.2. It represents the train booking process when the demographic variables are taken into consideration in the following order: $C \succ V \succ A \succ T$. The stage structure is given by: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3, w_4, w_5\}$, $u_4 = \{w_6\}$, $u_5 = \{w_7\}$, $u_6 = \{w_8\}$, $u_7 = \{w_9\}$.

The pair $\mathbb{G} = (\mathcal{T}, U)$ completely characterises the graphical structure of a CEG $\mathbb{C}$, where $\mathcal{T}$ is an event tree and U is the set of stages or the stage structure. So, the topology of a CEG is fully defined by its underlying stage tree. A triple $\mathbb{C} = (\mathcal{T}, U, \mathscr{P})$ formally defines a CEG model, where $\mathscr{P}$ is the elicited probabilistic measure corresponding to $\mathbb{G}$.

3.3 Conjugate Learning of CEGs using Dirichlet priors

Conjugate learning for CEGs (Thwaites et al., 2009, Smith, 2010, Freeman and Smith, 2011a) can handle data collected under an observational or random sampling design. Using analogous assumptions this method closely resembles the

standard and established learning framework developed for discrete Bayesian Networks (Heckerman, 1999, 2008). This assumes an event tree with $R+1$ situations and one of its possible CEG $\mathbb{C}$ having a set of $M+1$ stages $U = \{u_i : i = 0, \ldots, M\}$, where each situation in a stage u_i has L_i outgoing edges.

For the purpose of this thesis, a sample is said to be complete if it does not have missing values, errors in data entry and coding or inconsistent measurements. Now consider learning a CEG $\mathbb{C}$ from a complete sample $\boldsymbol{y} = \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_R\}$, where $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iL_i})$ is a vector summarising the number of units y_{ij} that transverse a situation s_i using each of its outgoing edge j in a event tree with $R+1$ situations. We can express this sample in a more appropriate way for CEG learning by defining $\boldsymbol{x} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_M\}$, where

$$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iL_i}) = \sum_{s_j \in u_i} \boldsymbol{y}_{s_j}.$$

In this formulation, x_{ij} indicates the number of units that transverse the stage u_i using its outgoing j. Note that $\boldsymbol{x}$ is defined according to the stages of a CEG whilst $\boldsymbol{y}$ is determined by the situations in the event tree. Let $\boldsymbol{\pi}_i = (\pi_{i1}, \ldots, \pi_{iL_i})$ be a probability vector corresponding to each stage u_i, where π_{ij} is the conditional probability of a unit in stage u_i unfolds through the emanating edge j of u_i.

Provided that the sampling experiment was properly randomised, the standard probability theory (Feller, 1971a) ensures that the likelihood is a multinomial likelihood defined by the parameters $\boldsymbol{\pi}_i$, $i = 0, \ldots, M$. It then takes the separable formulae given by a product of the likelihood of probability vectors corresponding to each stages in $\mathbb{U}$ as follows

$$L(\boldsymbol{\pi}) = \prod_{i=1}^{M} L_i(\boldsymbol{\pi}_i) = \prod_{i=1}^{M} \frac{\Gamma(\bar{x}_i + 1)}{\prod_{j=1}^{L_i} \Gamma(x_{ij} + 1)} \prod_{j=1}^{L_i} \pi_{ij}^{x_{ij}}. \tag{3.2}$$

where $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_M)$.

Next assume that the probability vectors $\boldsymbol{\pi}_i, i = 1, \ldots, M$, are mutually independent a priori. This property is called stage independence here. We note that when a CEG is also a BN then this assumption corresponds to the almost ubiquitous

assumption of global and local independences made for BN learning. Further assume as we also usually do for a BN model that each of these probability vectors has a Dirichlet prior distribution $Dir(\boldsymbol{\alpha}_i)$, where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{iL_i})$. Explicitly, this means that the prior distribution for $\boldsymbol{\pi}$ can be written as:

$$p(\boldsymbol{\pi}|\mathbb{C}) = \prod_{i=1}^{M} \frac{\Gamma(\bar{\alpha}_i)}{\prod_{j=1}^{L_i} \Gamma(\alpha_{ij})} \prod_{j=1}^{L_i} \pi_{ij}^{\alpha_{ij}}, \tag{3.3}$$

where $\alpha_{ij} > 0$, for all $i = 0, \ldots, M$ and $j = 1, \ldots, L_i$.

The Dirichlet prior distribution (equation 3.3) can then be updated given the multinomial likelihood (equation 3.2) to obtain the posterior distribution of the parameter vector $\boldsymbol{\pi}$ using Bayes formula as follows

$$\begin{aligned} p(\boldsymbol{\pi}|\boldsymbol{x}, \mathbb{C}) &\propto p(\boldsymbol{x}|\boldsymbol{\pi}, \mathbb{C}) p(\boldsymbol{\pi}) = \prod_{i=1}^{M} \frac{\Gamma(\bar{\alpha}_i)}{\prod_{j=1}^{L_i} \Gamma(\alpha_{ij})} \prod_{j=1}^{L_i} \pi_{ij}^{x_{ij}+\alpha_{ij}-1} \\ &= \prod_{i=1}^{M} p(\boldsymbol{\pi}_i|\boldsymbol{x}_i, \mathbb{C}) = \prod_{i=1}^{M} \frac{\Gamma(\bar{\alpha}^*_{ij})}{\prod_{j=1}^{L_i} \Gamma(\alpha^*_{ij})} \prod_{j=1}^{L_i} \pi_{ij}^{\alpha^*_{ij}-1}, \end{aligned} \tag{3.4}$$

where $\boldsymbol{\alpha}^*_i = \boldsymbol{\alpha}_i + \boldsymbol{x}_i$. Therefore, the posterior distribution has the same Dirichlet form as the prior distribution but with different parameters. Observe that this conjugate analysis as a CEG is extremely convenient since the parameter $\boldsymbol{\pi}_i$, $i = 1, \ldots, M$, associated with each stage u_i has a Dirichlet posterior distribution $Dir(\boldsymbol{\alpha}^*_i)$ and can then be learnt in closed form independently.

Another great advantage of a conjugate analysis is that the marginal likelihood of the corresponding model can be written in closed form. Thus, we have that

$$\begin{aligned} p(\boldsymbol{x}|\mathbb{C}) &= \int_{\boldsymbol{\pi}} p(\boldsymbol{x}|\boldsymbol{\pi}, \mathbb{C}) p(\boldsymbol{\pi}) \, \mathrm{d}\boldsymbol{\pi} = \int_{\boldsymbol{\pi}} \prod_{i=1}^{M} \frac{\Gamma(\sum_{j=1}^{L_i} \alpha_{ij})}{\prod_{j=1}^{L_i} \Gamma(\alpha_{ij})} \prod_{j=1}^{L_i} \pi_{ij}^{x_{ij}+\alpha_{ij}-1} \, \mathrm{d}\boldsymbol{\pi} \\ &= \prod_{i=1}^{M} \frac{\Gamma(\sum_{j=1}^{L_i} \alpha_{ij})}{\Gamma(\sum_{j=1}^{L_i} \alpha^*_{ij})} \prod_{j=1}^{L_i} \frac{\Gamma(\alpha^*_{ij})}{\Gamma(\alpha_{ij})}. \end{aligned} \tag{3.5}$$

The logarithmic form of the marginal likelihood can then be expressed as a *sum* over log-gamma functions of the hyper-parameters associated with the prior and posterior distributions. Explicitly, we then have that

$$\log p(\boldsymbol{x}|\mathbb{C}) = \sum_{i=1}^{M} \left\{ (a(\boldsymbol{\alpha}_i) - a(\boldsymbol{\alpha}^*_i)) - (b(\boldsymbol{\alpha}_i) - b(\boldsymbol{\alpha}^*_i)) \right\}, \tag{3.6}$$

where $a(\boldsymbol{\alpha}_p) = \log\Gamma(\bar{\alpha}_p)$ and $b(\boldsymbol{\alpha}_p) = \sum_{j=1}^{L_i} \log\Gamma(\alpha_{pj})$.

Notice here that although a log-gamma function is not a trivial mathematical function when manually computed there are various packages that calculate it precisely and in an extremely fast way. This enables us to explore the additive form of equation 3.6 in order to design efficient algorithm not only for learning but also for propagation and model search. For the purpose of this thesis it is then useful to introduce the definition of a standard CEG model.

Definition 20 (The Standard CEG model)**.** A CEG model learnt using a Bayesian framework is said to be a *standard CEG model* if the assumptions of complete random sampling, stage independence and Dirichlet prior distribution for each stage hold.

3.3.1 How to set up the prior distribution

As with all Bayesian methods we need to set up a sensible prior before we begin learning a model. In our context this translates into finding a way of appropriately setting the prior hyper-parameter $\boldsymbol{\alpha}$. This issue has already been addressed by a number of authors when learning a BN (Heckerman et al., 1995, Heckerman, 2008, Neapolitan, 2004, Cowell et al., 2007, Koller and Friedman, 2009, Korb and Nicholson, 2011). However here I will focus on just one that has been known to be particularly attractive to applied analyst in the analogous BN setting, see e.g. Heckerman et al. (1995), Heckerman (1999, 2008), Neapolitan (2004) and Koller and Friedman (2009).

The hyper-parameter $\boldsymbol{\alpha}$ in this family of Dirichlet prior distributions can be viewed as a phantom sample from the population, which is used to start the CEG learning process. Analogous to learn a BN, in this approach the analyst states that the strength of his prior judgement is equivalent to the phantom sample size $\bar{\alpha}$, i.e. the total number of phantom units $\bar{\alpha}$ which are supposed to pass through the root node of the CEG. Note that $\bar{\alpha}$ can be any positive real number.

Next the phantom sample size $\bar{\alpha}$ needs to be propagated over the CEG in order

to obtain the proper phantom sample that corresponds to the hyper-parameter vector $\boldsymbol{\alpha}$. This implies a conserving assumption that the total number of phantom units that transverse any position w_l is identical to the sum of all phantom units arriving at it. Let $pa(w_l)$ be the set of positions that are parent of w_l and $J_r(w_l)$ be the set of directed edges that emanate from a parent position $w_r \in pa(w_l)$ to the position w_l. Thus, for every position $w_l, w_l \in u_i$, the conserving condition can be explicitly expressed by

$$\bar{\beta}_l = \sum_{r \in pa(w_l)} \sum_{j \in J_r(w_l)} \beta_{rj} = \sum_{j=1}^{K_i} \beta_{ij}, \tag{3.7}$$

where $\boldsymbol{\beta}_l = (\beta_{l1}, \ldots, \beta_{lK_i})$ and where β_{lj} is the number of phantom units that arrive at position w_l and then takes the outgoing edge j.

Obviously, there are an explosive number of possible values that can be assumed for the explanatory hyper-parameter vectors associated of each stage. So to set them separately to initialise a model search algorithm would be a huge challenge. In real-world applications one usual way to circumvent this technical issue – and one I use here – is to further assume a uniform propagation of the phantom sample over the CEG. Although all routine methods have their drawbacks the assumption of uniform propagation has proved to be a relatively successful choice in analogous circumstances: see e.g. Heckerman et al. (1995), Heckerman (1999, 2008), Neapolitan (2004) and Koller and Friedman (2009). In the CEG framework this condition ensures that the total numbers of phantom units emanating from a position $w_l, w_l \in u_i$, through any two of its each outgoing edges are identical. Formally, this then means $\beta_{0j} = \frac{\bar{\alpha}}{L_0}, j = 1, \ldots, K_0$, and $\beta_{ij} = \frac{\bar{\beta}_i}{K_i}$, $i = 1, \ldots, K$, $j = 1, \ldots, K_i$. Finally, the a hyper-parameter $\boldsymbol{\alpha}_i$ is given by:

$$\boldsymbol{\alpha}_i = \sum_{w_l \in u_i} \boldsymbol{\beta}_l.$$

As the posterior hyper-parameter $\boldsymbol{\alpha}^*$ is obtained by a linear transformation of the prior hyper-parameter $\boldsymbol{\alpha}$ defined by the real sample $\boldsymbol{x}$, the prior mean and variance have a direct and simple link with the posteriors of these moments. In this sense, setting a small phantom sample size $\bar{\alpha}$ corresponds to adopting a weakly

informative prior distribution over each stage of our CEG model to start its learning process without inappropriately biasing it.

Under the conserving and uniform conditions, stages that are closer to the root position will have greater prior probability mass and less prior variance whilst stages near the sink position tend to have tinier prior probability mass and greater prior variance. Thus, the regularisation effect of the prior distribution over the learning process reduces as the stages is further distant from the root position. This is an attractive property since we a priori often expect to be less confident about processes unfolding in a more finer partition of the event spaces. Also observe that stages closer to the root node are often visited more frequently and so the possible stronger regularization impact of the prior distribution is counter-balanced by a greater real sample size collected on this stage.

To illustrate how to learn a CEG and to further discuss the setting of the prior distribution consider the example below about liver and kidney disorders.

Example 4 (Liver & Kidney disorders). *This toy example is an extended version of one presented in Thwaites et al. (2008).*

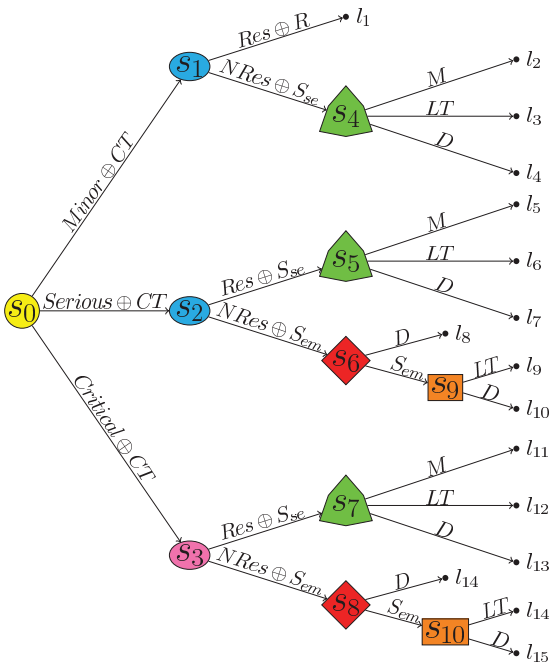


Figure 3.4: Staged Tree for liver and kidney disorders. The stage structure is given by: $u_0 = \{s_0\}, u_1 = \{s_1, s_2\}, u_2 = \{s_3\}, u_3 = \{s_4, s_5, s_7\}, u_4 = \{s_6, s_8\}, u_5 = \{s_9, s_{10}\}$.

A patient diagnosed with liver and kidney disorders is often refereed to a specialist by his general physician. Assume that this patient arriving at an specialised clinic is classified using three categories of dysfunction: minor (m), serious (s) or critical (c).

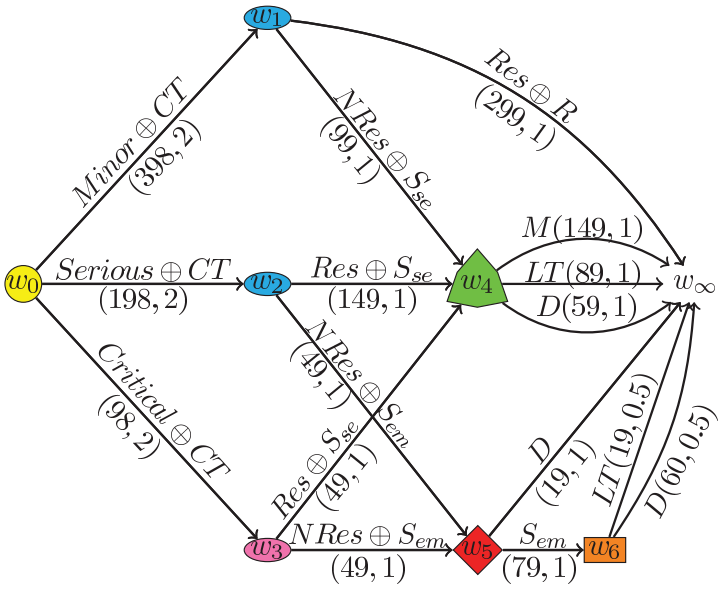


Figure 3.5: CEG for liver and kidney disorders supported by the stage tree in Figure 3.4. The pair (x_{ij}, β_{ij}) associated with an outgoing edge j from a position w_i corresponds to the number of patients (x_{ij}) in the sample and the value of β_{ij} yielded by the phantom sample. Stage structure: $U = \{u_0 = \{s_0\}, u_1 = \{s_1, s_2\}, u_2 = \{s_3\}, u_3 = \{s_4, s_5, s_7\}, u_4 = \{s_6, s_8\}, u_5 = \{s_9, s_{10}\}\}$. Position structure: $W = \{\{w_0 = \{s_0\}, w_1 = \{s_1\}, w_2 = \{s_2\}, w_3 = \{s_3\}, w_4 = \{s_4, s_5, s_7\}, w_5 = \{s_6, s_8\}, w_6 = \{s_9, s_{10}\}\}$.

Regardless of this classification every patient first receives an appropriate clinical treatment (CT). If the patient with a minor disorder responds (Res) to the treatment, he will be full recovery (R). Otherwise, he will be designated for a semi-elective surgery S_{se} whose results are lifetime monitoring (M), lifetime treatment (LT) or death (D). A patient with a serious or critical disorder who responds to the clinical treatment will be also designated to a semi-elective surgery S_{se}. However, if he does not respond $(NRes)$ to the first treatment and he is still alive, then he will be admitted to an emergency surgery S_{em} whose consequences are either a lifetime of treatment or death.

Assume that patients with minor or serious disorders have the same probability of responding to the clinical treatment. Also assume that the result from a semi-elective surgery has the same probability distribution for all patients. Finally assume that patients with a non-minor disorders have the same rate of survival along the clinical treatment and the emergency surgery. Figures 3.4 and 3.5 depict, respectively, the staged tree for this process and its corresponding CEG.

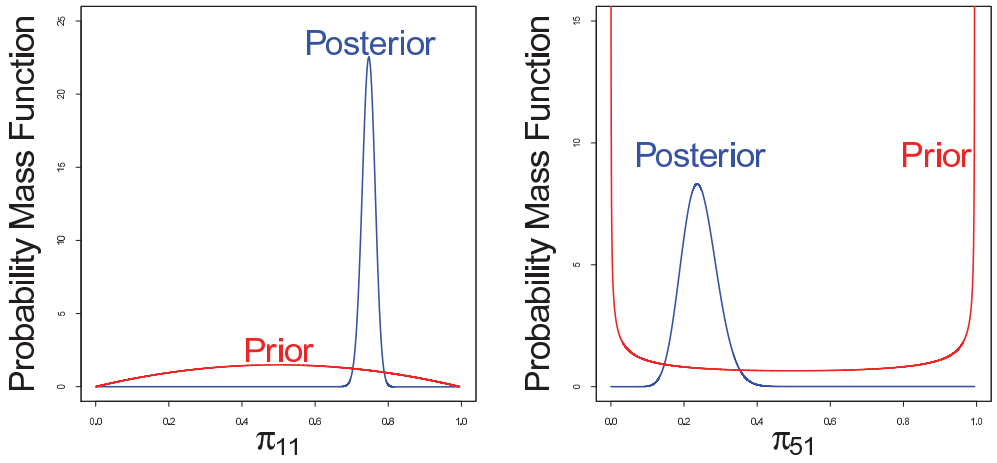
Now take a complete randomised trial with 694 *patients and set* $\bar{\alpha} = 6$. *The sample and the propagation of the* $\bar{\alpha}$ *are showed along the edges of the CEG in Figure 3.5. So,* $\boldsymbol{\alpha}_0 = (2, 2, 2)$. *Since there are two edges arriving in stage* $u_1 = \{w_1, w_2\}$, *we then have that* $\bar{\alpha}_1 = 4$ *and* $\boldsymbol{\alpha}_1 = (2, 2)$. *As stage* u_2 *is constituted by a single position, it follows directly that* $\bar{\alpha}_2 = 2$ *and* $\boldsymbol{\alpha}_2 = (1, 1)$. *The other hyper-parameters are fixed in a similar way. Table 3.4 shows the prior and posterior probabilities associated with each stage of this CEG.*

Stage	**State Space**	**Prior Distribution**	**Sample** $\boldsymbol{x}_i$	**Posterior Distribution**	**Posterior Mean (%)**
u_0	(m, s, c)	$Dir(2, 2, 2)$	$(398, 198, 98)$	$Dir(400, 200, 100)$	$(57, 29, 14)$
u_1	$(Res, NRes)$	$Dir(2, 2)$	$(448, 148)$	$Dir(450, 150)$	$(75, 25)$
u_2	$(Res, NRes)$	$Dir(1, 1)$	$(49, 49)$	$Dir(50, 50)$	$(50, 50)$
u_3	(M, LT, D)	$Dir(1, 1, 1)$	$(149, 89, 59)$	$Dir(150, 90, 60)$	$(50, 30, 20)$
u_4	(D, S_{em})	$Dir(1, 1)$	$(19, 79)$	$Dir(20, 80)$	$(20, 80)$
u_5	(LT, D)	$Dir(0.5, 0.5)$	$(19, 60)$	$Dir(19.5, 60.5)$	$(24, 76)$

Table 3.4: Prior and posterior probability distributions and data associated with each stage of the CEG depicted in Figure 3.5.

Figure 3.6 show the prior and posterior probability distributions of stages u_1 *and* u_5. *Table 3.5 presents their prior and posterior 95% credible intervals for these stages and their prior and posterior variances. We can see that the prior distributions have greater variance since their probability mass is fairly distributed over the parameter space. In fact the prior variances of each component of stages* u_1 *and* u_5 *are, respectively,* 0.05 *and* 0.125 *and their corresponding 95% credible intervals*

are, respectively, $(0.09, 0.91)$ *and* $(0.002, 0.998)$*. As we move from the root to the sink position in a CEG the stage prior distributions become more diffused whilst stages that are closer to the root position have some concentration of mass of probability around the value of what it would be a uniform distribution:* 0.5 *for stage* u_1*.*



(a) Stage u_1:
$\pi_{11}(prior) \sim Beta(2, 2)$
$\pi_{11}(posterior) \sim Beta(450, 150)$

(b) Stage u_5:
$\pi_{51}(prior) \sim Beta(0.5, 0.5)$
$\pi_{51}(posterior) \sim Beta(19.5, 60.5)$

Figure 3.6: Prior (red) and posterior (blue) probability distributions of stages u_1 and u_5 associated with CEG depicted in Figure 3.5. These probability distributions are also showed in Table 3.4.

However stages that closer to the root position are often visited more frequently and so tend to have greater concentration of mass probability a posteriori. This is sufficient to counter-balance any possible prior bias towards uniformity that a prior can impose over these stages as we can observe in Figure 3.6a. The posterior variance of stage u_1 *is less than* $1,0 \times 10^{-3}$ *and their 95% credible posterior interval is narrow (gap of* 0.07*).*

In contrast, stage u_5 *corresponds to position* w_6*. Since this is the most distant position from the root node in our CEG we can expect that this stage is associated with a smaller sample size than most of the other stages. This is actually what happens in our example. It therefore has a higher variance* $(2,3 \times 10^{-3})$ *and a*

wider 95% credible posterior intervals (gap of 0.18). However even in this case the posterior distribution provides us with a reliable and clear picture of our process despite the sample size being only 79 out of the 694 individuals that constitute our overall sample. See also Figure 3.6b.

Probability Distribution	**State Space**	**Mean (95% Credible Interval) (%)**			**Variance ($\times 10^{-4}$)**
u_0 - Prior	(m, s, c)	$33(5, 72)$	$33(5, 72)$	$33(5, 72)$	$(317, 317, 317)$
u_0 - Posterior	(m, s, c)	$57(53, 61)$	$29(25, 32)$	$14(12, 17)$	$(3.4, 2.9, 1.7)$
u_1 - Prior	$(Res, NRes)$	$50(9, 91)$	–	$50(9, 91)$	$(500, 500)$
u_1 - Posterior	$(Res, NRes)$	$75(71, 78)$	–	$25(22, 29)$	$(3.1, 3.1)$
u_5 - Prior	(LT, D)	$50(0.2, 99.8)$	–	$50(0.2, 99.8)$	$(1250, 1250)$
u_5 - Posterior	(LT, D)	$24(16, 34)$	–	$76(66, 84)$	$(23, 23)$

Table 3.5: Mean, 95% Credible Interval and Variance corresponding to the prior and posterior probability distributions of stages u_0, u_1 and u_5 associated with CEG depicted in Figure 3.5. In the column State Space it is showed the categories of the random variable associated with each stage. For a particular category, the lower and upper bounds of the 95% credible interval are given in parenthesis next to the mean. If a stage only has two categories, the middle column of the field Mean is filled with $-$. The column variance depicts the variance associated with the categories of each random variable.

□

3.4 Propagating information using uncoloured graphs

Propagating information is a process of calculating and updating the probability distributions associated with a graphical model conditioning upon the new information that becomes available (Cowell et al., 2007, Korb and Nicholson, 2011). In this section I will first discuss the algorithm developed by Thwaites et al. (2008) to

propagate information over a CEG whose conditional probability tables are known. I note that this algorithm is analogous to one developed for retraction of evidence in a BN (Cowell and Dawid, 1992). I will then present a new modified version of that algorithm which is actually computationally more efficient than the original one. The method for propagating evidence is illustrated using the liver and kidney example introduced in Section 3.3.1.

3.4.1 The Standard framework for propagating evidence over a CEG

Of course, not all kind of evidence can be propagated through a CEG $\mathbb{C} = (V, E)$. This also happens in the BN framework where evidence to be propagated has to be constrained to that which can be defined in terms of some subset of the sample space of variables used to construct the BN model. In this case, passing information based on an arbitrary function of the variables using local massages can disrupt the conditional independence structures upon which the BN propagation algorithm relies on to perform the local updates.

Analogously to the BN algorithm we assume that the information $\mathcal{I}$ in $\mathbb{C}$ is identifiable in terms of some subset of the sample spaces of the random variables associated with the set of position V in $\mathbb{C}$. A set of evidence satisfying this property is said to constitute a $\mathbb{C}$-compatible information. Formally, $\mathcal{I}$ is $\mathbb{C}$-*compatible* if and only if there exists a minimal set of edges $E(\mathcal{I}), E(\mathcal{I}) \subseteq E$, such that $\mathcal{I}$ can be expressed as a set of paths

$$\Lambda(\mathcal{I}) = \{\lambda \subseteq \mathbb{C};\ \text{every edge of a path } \lambda \text{ is in } E(\mathcal{I})\}.$$

Information is said to be *trivial* $\mathbb{C}$-*compatible* if $E(\mathcal{I})$ includes all edges of E, i.e. $E(\mathcal{I}) = E$.

Note that the CEG provides us with a more flexible framework for propagating evidence than a BN whose information to be retrieved must be defined in terms of a predetermined set of random variables. So the types of information that are compatible with CEG propagation are more varied than those compatible with a BN. In fact, a CEG is able to propagate information that would destroy the junction

tree framework of a BN even when the CEG is a re-written version of a BN. I next will examine the three standard steps required to propagated a compatible information over a CEG: construction of the transporter CEG, evidence collection and evidence distribution.

Obtaining the transporter CEG

Propagating a new information using a BN model $\mathbb{B}$ requires us a pre-processing step. At this stage, the corresponding DAG of $\mathbb{B}$ needs to be moralised, triangulated and transformed in an junction tree whose vertices are cliques of variables. This enables us to identify the most relevant conditional independence structures embedded into the model and translate these into appropriate Markov properties for belief propagation. Recall that in a junction tree a variable can take part in different cliques and so can be associated with two or more vertices in the junction tree.

In a similar way, the CEG $\mathbb{C} = (V, E)$ has to be prepared for belief propagation. Being naturally supported by a tree the pre-processing is rather simple and consists only in taking out the colours of the initial CEG graph. This uncoloured graph $\mathbb{C}_u = (V, E)$ is called the transporter of $\mathbb{C}$. Note that both graphs, $\mathbb{C}_u$ and $\mathbb{C}$, have the same graphical topology except that $\mathbb{C}_u$ lose track of those positions that are at the same stage. This means that a position plays a similar role of cliques in the BN framework. It also implies that floret conditional independence structures associated with a pair of positions w_a and w_b not holding over their unfolding event trees are ignored. However, positions are not merged and so their original conditional probability tables are not associated with different vertices in the transporter CEG.

Given a $\mathbb{C}$-compatible information and obtained a transporter CEG $\mathbb{C}_u$ we can update our belief in two further steps. For this purpose, assume that the sets of positions and edges in $\mathbb{C}$ are well-ordered in the sense if $i_1 < i_2$, then neither a position w_{i_1} nor an edge e_{i_1} lie downstream, respectively, of a position w_{i_2} and an

edge e_{i_2} of any root-to-sink path in $\mathbb{C}$.

Evidence Collection

In this step each position absorbs the new information using backward strategy: from the sink position w_∞ to the root position w_0. The collected evidence is stored in each position w by a new pair $\{\boldsymbol{\tau}(w), \phi(w)\}$, where $\boldsymbol{\tau}(w) = (\tau_1(w), \ldots, \tau_{K_w}(w))$. Here $\tau_i(w)$ is a *potential* corresponding to each propagated evidence arriving at w through its outgoing edge i and $\phi(w)$ is an *emphasis* that merges additively all potentials arriving at w. The potential component $\tau_i(w)$ associated with the i^{th} outgoing edge $e_i(w)$ of w is given by

$$\tau_i(w) = \begin{cases} 0 & \text{if } e_i(w) \notin \Lambda(\mathcal{I}), \\ \pi_{wi} & \text{if } e_i(w) \in \Lambda(\mathcal{I}). \end{cases} \tag{3.8}$$

and

$$\phi(w) = \sum_{i=1}^{K_w} \tau_i(w). \tag{3.9}$$

A position w is said to be *accommodated* whenever its pair $\{\boldsymbol{\tau}(w), \phi(w)\}$ is calculated. In this stage all position should be accommodated in its reverse order excluding the sink position w_∞, where $w_\infty \equiv w_{|V|}$; i.e. from $w_{|V|-1}$ to w_0.

Evidence Distribution

In the last step all collected evidence is distributed forwards in the transporter CEG $\mathbb{C}_u$. This operation delivers the revised probabilities $\hat{\boldsymbol{\pi}} = (\hat{\boldsymbol{\pi}}_{w_0}, \ldots, \hat{\boldsymbol{\pi}}_{w_\infty})$ conditional on the information $\mathcal{I}$. So, for all $w \in V$, from w_0 to $w_{|V|-1}$, set:

$$\hat{\boldsymbol{\pi}}_w = \begin{cases} \mathbf{0} & \text{if } e_i(w) \notin \Lambda(\mathcal{I}), \\ \frac{\boldsymbol{\tau}(w)}{\phi(w)} & \text{if } e_i(w) \in \Lambda(\mathcal{I}). \end{cases} \tag{3.10}$$

For any position w in the resulting CEG $\hat{\mathbb{C}}$ the updated probability of arriving at w along a root-to-w path $\lambda(w_0, w) = (V(\lambda), E(\lambda))$ is given by:

$$p(w|\lambda) = \prod_{l=0}^{|E(\lambda)|-1} \hat{\pi}_{w_{i_l} e_{i_{l+1}}}, \tag{3.11}$$

where:

$$V[\lambda] = \{w_{i_l}; l = 0, \ldots, |E(\lambda)|, w_{i_0} \equiv w_0, w_{i_{|E(\lambda)|}} \equiv w\},$$

$$E[\lambda] = \{e_{i_l}; l = 1, \ldots, |E(\lambda)|\}, \text{ and}$$

$$\lambda(w_0, w) = (w_{i_0}, e_{i_1}[\lambda], w_{i_1}[\lambda], \ldots, e_{i_{|E(\lambda)|}}[\lambda], w_{i_{|E(\lambda)|}}[\lambda]).$$

It therefore follows that the updated probability to arrive at position w has the following form:

$$p(w) = \sum_{\lambda \in \Lambda(w)} \prod_{l=0}^{|E(\lambda)|-1} \hat{\pi}_{w_{i_l} e_{i_{l+1}}}, \tag{3.12}$$

where $\Lambda(w)$ is the set of all w_0-to-w paths. In analogy to the BN propagation algorithm (Cowell and Dawid, 1992, Equation 6), each atom of the probability space of $\hat{\mathbb{C}}$, which corresponds to a w_0-to-w_∞ path λ, $\lambda \in \Lambda(w_\infty)$, has a probability mass defined by the invariance formula:

$$\hat{\pi}(\lambda) = p(w_\infty|\lambda) = \prod_{l=0}^{|E(\lambda)|-1} \hat{\pi}_{w_{i_l} e_{i_{l+1}}} = \frac{\prod_{l=0}^{|E(\lambda)|-1} \tau_{e_{i_l}}(w_{i_l})}{\prod_{l=0}^{|E(\lambda)|-1} \phi(w_{i_l})}. \tag{3.13}$$

From equation 3.13 we can see that the computational cost of performing inference can be substantially reduced if a reduced CEG, whose vertices and edges are just those with non-zero emphasis and potential in $\hat{\mathbb{C}}$, is used. Observe that any non-trivial $\mathbb{C}$-compatible information strictly enables us to reduce the number of edges in $\hat{\mathbb{C}}$. For further discussion about computational efficiency of this algorithm compared to that for BNs, see Thwaites et al. (2008).

3.4.2 Modified version of the propagation algorithm

The original algorithm (Thwaites et al., 2008) propagates new information in the three *different* steps described above. However it is straightforward to see that these three phases can be performed simultaneously during the computational calculations. This constitutes my contribution for this method because it enables us to write a more efficient code but nevertheless does not change the theoretical properties of the original version. Therefore the proof for this new version is completely identical to that one initially introduced in Thwaites et al. (2008).

A pseudo-code of the propagation algorithm is presented below. Note that the order of the operations is completely defined by the topology of the original CEG $\mathbb{C}$ and so can be set beforehand.

As in the earlier version, the resulting reduced CEG $\hat{\mathbb{C}}$ from this algorithm is not necessarily minimal: there might exist two different vertices whose corresponding unfolding staged trees are graphically and probabilistically isomorphic under the updated probability distribution $\hat{\boldsymbol{\pi}}_w$. Thus, these vertices would actually be in the same position and could be further merged into a single vertex.

Also note that the graph can be coloured according to the updated probabilities $\hat{\boldsymbol{\pi}}_w$. Obviously, it is possible to obtain a minimal coloured CEG if an additional step is included in the algorithm for this purpose. Nevertheless this is not strictly necessary for evidence propagation and the interpretation of the results.

Algorithm 1: The propagation algorithm

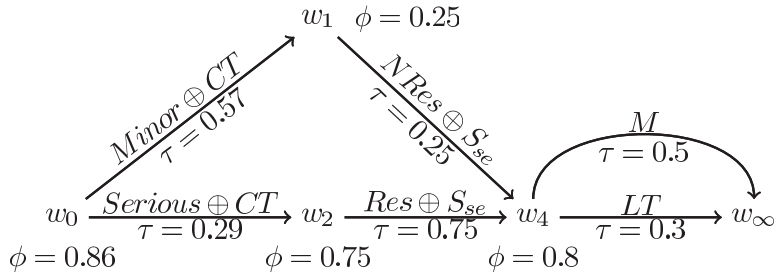
Input: A well-ordered CEG $\mathbb{C} = (V, E)$ and a $\mathbb{C}$-compatible information $\mathcal{I}$.

Output: An uncoloured CEG $\hat{\mathbb{C}} = (\hat{V}, \hat{E})$.

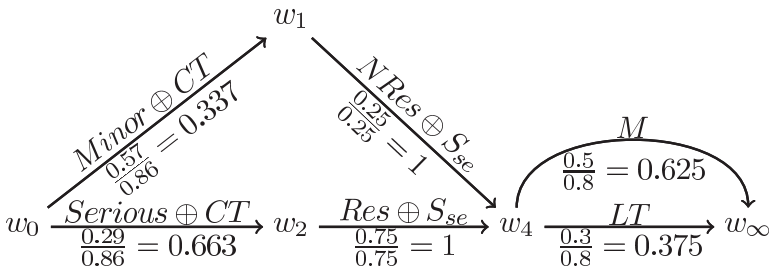
Set $\hat{V} = \emptyset$, $\hat{E} = \emptyset$, and $\phi = 0$.
Initialise $\hat{\boldsymbol{\pi}}$ such that $|\hat{\boldsymbol{\pi}}| = |V - 1|$.
for *j in* $|V-1| \rightarrow 0$ **do**
    Initialise a vector $\boldsymbol{\tau} = -\mathbf{1}$ such that $|\boldsymbol{\tau}| = K_{w_j}$
    for *i in* $1 : K_{w_j}$ **do**
        $\tau_i \leftarrow \tau_i(w_j)$ (Equation 3.8).
        if $\tau_i(w_j) \neq 0$ **then**
            $\hat{E} \leftarrow \hat{E} \cup \{e_i[w_j]\}$
    $\phi \leftarrow \phi(w_j)$ (Equation 3.9)
    if $\phi(w_j) \neq 0$ **then**
        $\hat{V} \leftarrow \hat{V} \cup \{w_j\}$
    $\hat{\boldsymbol{\pi}}_i \leftarrow \frac{\boldsymbol{\tau}}{\phi}$
return $\hat{V}, \hat{E}, \hat{\boldsymbol{\pi}}$

3.4.3 Example

Recall Example 4. Suppose that a patient who has worked overseas had a liver and kidney disorder during the last year. Returning to his home country, he went to a specialised clinic and he reported to a physician that his case was diagnosed as non-critical but a semi-elective surgery was prescribed to him. Before requiring further clinical exams, the physician would like to assess his actual health state using the data collected during the clinical interview. For this purpose, he used a clinical decision-making support software based on the CEG in Figure 3.5 whose conditional probabilities were fixed to have the posterior mean given by Table 3.4.



(a) Transporter CEG $\mathbb{C}_u$ with the potential τ associated with each edge and the emphasis ϕ corresponding to each vertex.



(b) The Updated CEG $\hat{\mathbb{C}}$. Each updated non-zero probability is showed below its corresponding edge.

Figure 3.7: The transporter and updated CEGs associated with the CEG depicted in Figure 3.5 when it is known that a patient is classified as non-critical and had been submitted to a semi-elective surgery.

In this case the available information corresponds to the set of edges

$$\mathcal{I} = \{(w_0, w_1), (w_0, w_2), (w_1, w_4), (w_2, w_4), (w_4, w_\infty, M), (w_4, w_\infty, LT)\},$$

where (w_4, w_∞, l) denotes the edge (w_4, w_∞) with label l. After introducing the information $\mathcal{I}$ in the software the physician obtains the updated CEG $\hat{\mathbb{C}}$ in Figure 3.7b. Figure 3.7a shows the transporter CEG used to propagated the information with the corresponding potentials and emphases. For this patient, the odds between a minor or a serious disorder is then $2:1$ whilst the odds between monitoring and lifetime medication is $5:3$.

Note that the conditional probabilities associated with this CEG can be stored using only 14 cells. In contrast, it is not so efficient to represent this using a BN model. This is not only because of the context-specific statement but also because the process develops in a highly asymmetric way. A possible 5-variable BN model for this process is presented in Figure 3.8, where:

1. Variable X_1 denotes the type of the disorders (minor, serious, critical).
2. Variable X_2 distinguishes whether a patient responds to the clinical treatment.
3. Variable X_3 differentiates among the types of surgery (none, semi-elective, emergency).
4. Variable X_4 flags if the patient is alive.
5. Variable X_5 describes the final outcome (full recovery, monitoring, medication, death) associated with a surgery.

The responses to the first clinical treatment, which are stored by the set of positions $\{w_1, w_2, w_3\}$ in the CEG, are now represented through the variables X_2, X_3 and X_5. The variable X_5 also stores the information on the result of a surgery, which corresponds to the set of positions $\{w_1, w_4, w_6\}$ in the CEG. The BN model now requires 42 cells -18 for the clique $\{X_1, X_2, X_3\}$ and 24 for the clique $\{X_3, X_4, X_5\}$- to store the conditional probabilities, 28 of which are storing the value zero -12 in the clique $\{X_1, X_2, X_3\}$ and 16 in the clique $\{X_3, X_4, X_5\}$. This implies a greater computational cost for propagating evidence than using a CEG model that

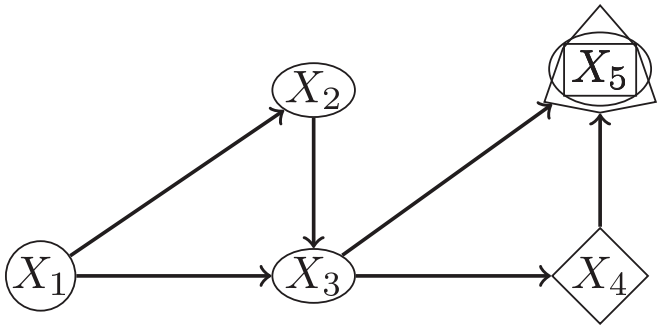


Figure 3.8: BN corresponds to Example 4. The geometric shape of each vertex corresponds to the geometric shapes of positions that convey the same kind of information in the CEG depicted in Figure 3.5.

provides us a more parsimonious storing structure. Moreover, using a BN model the preprocessing phase, the collection of evidence and its distribution constitute three different steps. In fact, the last two steps often cannot be conducted in a single step in a BN model because a variable can be in two different cliques of the junction tree.

The propagation of evidence using CEGs is often much more efficient than BNs, particularly in non-product-space settings where the process develops asymmetrically. A CEG graph can embody such asymmetry by having w_0-to-w_∞ paths with different lengths. It can thus provide a framework enabling us to avoid performing repetitive calculations and propagating zeros in sparse but large probability tables. As we have demonstrated in the simple example above, this would not be possible if we used a BN model with a naive propagation algorithm in this sort of context. Moreover, the range of compatible information with a CEG model is larger and more general that using a BN model. This allows us to perform more detailed analysis and use a finer information partition, which is especially important when data is contingently censored.

Chapter 4

Standard Bayesian CEG Model Selection

In this Chapter algorithms for Bayesian CEG model selection are extensively discussed. Here I report some new contributions to the literature of CEG structural learning. I will extend the Dirichlet characterisation for a set of CEGs supported by a single event tree (Freeman and Smith, 2011a) to a set of CEGs yielded by a collection of event trees (Section 4.1.3). I will also introduce some technical novelties that will make it possible to scale up the current model search algorithms. These are based on a new construction called a hyper-stage (Section 4.2.1). I further provide advances in the algorithmic structures (Sections 4.2.1 and 4.4) that are shown to be compatible with parallel computing (Section 4.4). A first ever analysis of the train booking process described in Section 3.1 is presented in Section 4.5.

One of the most challenging characteristics of CEG model selection is that the associated space of models is immense. To search the CEG model we therefore need to design efficient algorithms. Standard CEGs provide us with a simple and well-established way to do this because the posterior probability of a model can be written down analytically and in closed form. So, I first will review the conditions that guarantee that all models in the model space yielded by only one event tree are standard CEGs (Freeman and Smith, 2011a).

At this point, I will present a particularly useful class of CEG models called the Stratified CEG (SCEG) (Cowell and Smith, 2014). SCEG models constitute an important CEG class because it contains all discrete context-specific BNs as a special case. It also enables us to explore plausible collections of causal hypotheses. We will see that the SCEG model space is often yielded by many different event trees. Analogously to the conditions for BN model search (Heckerman and Geiger, 1995, Geiger and Heckerman, 1997, Heckerman, 1999) I will then introduce two *new* assumptions. These guarantee that *all* models in a SCEG model space are standard CEGs. I will also review some possible ways to set a prior distribution over the CEG model space.

I will then proceed to explore two different strategies to search over the CEG model space: a greedy CEG model search algorithm and a dynamic programming model search algorithm. I will also consider two different modelling situations faced by technical analysts in practice, namely when the event tree is completely defined by domain knowledge a priori and when it is not available. In this latter situation, the dynamic programming method is able to conduct a search of the SCEG model space that is guaranteed to find an optimal scoring model.

Despite this advance the size of the CEG model space can still represent a serious challenge for model selection over the space of CEG models whose probability space has a moderate or large number of primitive probabilities. Then a full search method quickly becomes unfeasible as the number of these explanatory variable increases. Silander and Leong (2013) and Cowell and Smith (2014)) both recognised that "greedy" search algorithms would often be required when the size of the model space was scaled up. These methods find the best of a class of a priori promising models.

One such heuristic approach is the Agglomerative Hierarchical Clustering (AHC) algorithm. Freeman and Smith (2011a) customised this strategy for CEG model selection when the event tree is completely specified. This algorithm can be used for any kind of family of CEGs. This is in contrast to the dynamic programming

algorithm that only applies for SCEG models. After a review of this algorithm, I will develop a *new* concept called *hyper-stage*. This will allow me to propose a more computationally efficient AHC algorithm. I will also show that this concept also enables us to embed different explanatory and causal hypotheses within the CEG model search based on domain information.

I will next outline some challenges of searching the CEG model space, as well as some *new* strategies to address these challenges. Finally I use these CEG model search algorithms to understand for the first time the train booking process described in Section 3.1.

4.1 Bayes Factors and CEG model selection

To perform model selection, it is necessary to first define a family of event trees $\mathcal{T}$ that spans our CEG model space $\boldsymbol{\mathcal{C}}$. This will constitute the model search space. We next need to specify a score that will be used to compare these models. There are a variety of methods and here we adopt the Bayesian paradigm where each model $\mathbb{C}, \mathbb{C} \in \boldsymbol{\mathcal{C}}$, is scored by its log posterior probability $p(\mathbb{C}|\boldsymbol{x})$. For example, one objective may be to find the CEG that maximizes this score, the maximum a posteriori CEG (MAP CEG). Setting a prior probability $q(\mathbb{C})$ for each model $\mathbb{C}$ in $\boldsymbol{\mathcal{C}}$, the log posterior probability of $\mathbb{C}$ is given by

$$Q(\mathbb{C}) \;\; = \;\; \log p(\mathbb{C}|\boldsymbol{x}) \propto \log p(\boldsymbol{x}|\mathbb{C}) q(\mathbb{C}) \tag{4.1}$$

Assuming that the conditions leading to a standard CEG model hold we can then use equation 3.6 and write $Q(\mathbb{C})$ analytically as

$$Q(\mathbb{C}) = \sum_{i=1}^{M} \{(a(\boldsymbol{\alpha}_i) - a(\boldsymbol{\alpha}_i^*)) - (b(\boldsymbol{\alpha}_i) - b(\boldsymbol{\alpha}_i^*))\} + \log q(\mathbb{C}) + K, \tag{4.2}$$

where $K = \log p(\boldsymbol{x})$ is the normalization constant. The log posterior Bayes Factor between two models $\mathbb{C}_1$ and $\mathbb{C}_2$ can then be written in closed form as

$$
\begin{aligned}
\log pBF(\mathbb{C}_1, \mathbb{C}_2) &= \log \frac{p(\mathbb{C}_1|\boldsymbol{x})}{p(\mathbb{C}_2|\boldsymbol{x})} \\
&= \log q(\mathbb{C}_1) - \log q(\mathbb{C}_2) + \log p(\boldsymbol{x}|\mathbb{C}_1) - \log p(\boldsymbol{x}|\mathbb{C}_2) \\
&= \log q(\mathbb{C}_1) - \log q(\mathbb{C}_2) \\
&+ \sum_{i=1}^{L_1} \{(a(\boldsymbol{\alpha}_{\mathbf{1}i}) - a(\boldsymbol{\alpha}^*_{\mathbf{1}i})) - (b(\boldsymbol{\alpha}_{\mathbf{1}i}) - b(\boldsymbol{\alpha}^*_{\mathbf{1}i}))\} \\
&- \sum_{i=1}^{L_2} \{(a(\boldsymbol{\alpha}_{\mathbf{2}i}) - a(\boldsymbol{\alpha}^*_{\mathbf{2}i})) - (b(\boldsymbol{\alpha}_{\mathbf{2}i}) - b(\boldsymbol{\alpha}^*_{\mathbf{2}i}))\}. \qquad (4.3)
\end{aligned}
$$

Therefore to search a CEG model space using a Bayesian framework we need to choose a prior probability over the model space ($q(\mathbb{C})$) and over the parameter space of each model (hyper-parameter $\boldsymbol{\alpha}$). Before analysing each of these points separately I will present a useful family of CEGs.

4.1.1 A Stratified Chain Event Graph

A useful class of CEGs for model selection is the so-called Stratified CEGs (SCEGs) (Cowell and Smith, 2014). The SCEG class has the discrete BNs and context-specific BNs as particular subclasses of models. As in the BN framework, a SCEG is defined by a set of random variables $\boldsymbol{\mathcal{Z}} = \{Z_1, Z_2, \ldots, Z_N\}, N \geq 2$, where each variable $Z_n, n = 1, \ldots, N$, corresponds to a particular measurement on each of the units observed in any target system. Let I be a permutation of the set $\{1, 2, \ldots, N\}$ such that

$$\{1, 2, \ldots, N\} \overset{I}{\longmapsto} (i_1, i_2, \ldots, i_N),$$

which is used to order the set of variables $\boldsymbol{\mathcal{Z}}$ as following

$$\boldsymbol{\mathcal{Z}} \overset{I}{\longmapsto} (Z_{i_1}, Z_{i_2}, \ldots, Z_{i_N}) \triangleq \boldsymbol{Z}(I),$$

where $\boldsymbol{Z}(I)$ is the ordered sequence of the variables in $\boldsymbol{\mathcal{Z}}$ spanned by a permutation I. Now let

$$\mathbb{Z}^{(k)}(I) = \mathbb{Z}_{i_1} \times \mathbb{Z}_{i_2} \times \cdots \times \mathbb{Z}_{i_k}$$

be the product space of the first k variables in $\boldsymbol{Z}(I)$.

Each permutation I spans a different event tree $\mathcal{T}(\boldsymbol{Z}(I))$ called $\boldsymbol{\mathcal{Z}}-$compatible event tree (Definition 21). The set of all possible $\boldsymbol{\mathcal{Z}}-$compatible event trees

constitute the family of event trees denoted by $\boldsymbol{\mathcal{T}}_{\boldsymbol{\mathcal{Z}}}$. This family $\boldsymbol{\mathcal{T}}_{\boldsymbol{\mathcal{Z}}}$ supports the SCEG class.

Definition 21 ($\boldsymbol{\mathcal{Z}}-$compatible Event Tree)**.** An event tree $\mathcal{T}(\boldsymbol{Z}(I))$ is said to be $\boldsymbol{\mathcal{Z}}-$*compatible* if the following two conditions hold:

1. Its vertex set $V(\mathcal{T}(\boldsymbol{Z}(I)))$ consists of a root vertex s_0 together with a set of vertices $s(\boldsymbol{z}^{(k)})$, one for each $\boldsymbol{z}^{(k)} = (z_{i_1}, z_{i_2}, \ldots, z_{i_k})$ in $\mathbb{Z}^{(k)}(I)$, and $1 \leq k \leq N$. Note that each $s(\boldsymbol{z}^{(N)})$ is a leaf node.
2. Its edge set $E(\mathcal{T}(\boldsymbol{Z}(I)))$ is formed by the set of labelled edges $(s_0, s(\boldsymbol{z}^{(1)}), z_{i_1})$, $z_{i_1} \in \mathbb{Z}_{i_1}$, together with a set of labelled edges $(s(\boldsymbol{z}^{(k)}), s(\boldsymbol{z}^{(k+1)}), z_{i_{k+1}})$, where $\boldsymbol{z}^{(k+1)} = (\boldsymbol{z}^{(k)}, z_{i_{k+1}})$ and $z_{i_{k+1}} \in \mathbb{Z}_{i_{k+1}}$, one for each $\boldsymbol{z}^{(k)}$ in $\mathbb{Z}^{(k)}(I)$, and $1 \leq k \leq N-1$.

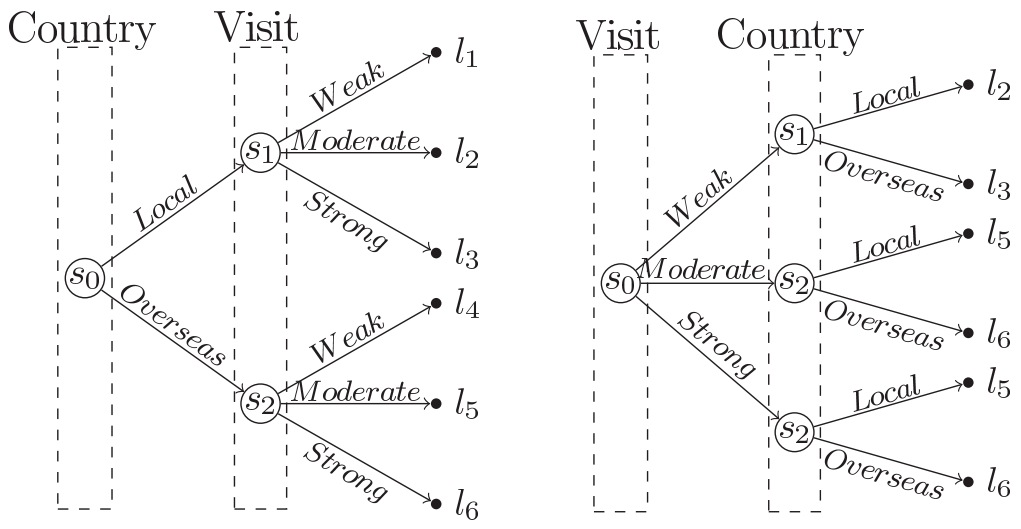
Observe that in any event tree $\mathcal{T}(\boldsymbol{Z}(I)), \mathcal{T}(\boldsymbol{Z}(I)) \in \boldsymbol{\mathcal{T}}_{\boldsymbol{\mathcal{Z}}}$, all of its non-root vertices $s(\boldsymbol{z}^{(k)})$, $\boldsymbol{z}^{(k)} \in \mathbb{Z}^{(k)}(I)$, are at the same distance k from the root v_0. Also note that each edge $(s(\boldsymbol{z}^{(k)}), s(\boldsymbol{z}^{(k+1)}), z_{i_{k+1}})$ is labelled by a possible value $z_{i_{k+1}} \in \mathbb{Z}_{i_{k+1}}$ of the variable $Z_{i_{k+1}}$ on the ordered components of $\boldsymbol{Z}(I)$ determined by I. For further discussion about $\boldsymbol{\mathcal{Z}}-$compatible Event Trees, see Example 5.

Example 5 (Train Booking with two demographic variables). *Recall the train booking process presented in Section 3.1. For simplicity, assume now that we would like to explore the interplay between the demographic variables Country (C) and Visit (V).*

Two demographic variables yield only two possible $\boldsymbol{\mathcal{Z}}-$compatible Event Trees, where $\boldsymbol{\mathcal{Z}} = \{C, V\}$. These event trees are depicted in Figure 4.1. They correspond to the variables orders $\boldsymbol{Z}(I_1) = (C, V)$ and $\boldsymbol{Z}(I_2) = (V, C)$. Note that $\boldsymbol{\mathcal{T}}_{\boldsymbol{\mathcal{Z}}} = \{\mathcal{T}(\boldsymbol{Z}(I_1)), \mathcal{T}(\boldsymbol{Z}(I_2))\}$. □

Note that if a process is characterised by asymmetrical developments and so has a non-product event space, then its corresponding event tree will often be not $\boldsymbol{\mathcal{Z}}$-compatible. We are now able to define the SCEG class.

Definition 22 (Stratified Chain Event Graph)**.** A CEG is called a $\boldsymbol{\mathcal{Z}}$*-Stratified Chain Event Graph* ($\boldsymbol{\mathcal{Z}}$-SCEG) if and only if its event tree is a $\boldsymbol{\mathcal{Z}}$-compatible event



(a) Event Tree $\mathcal{T}(\boldsymbol{Z}(I_1))$

(b) Event Tree $\mathcal{T}(\boldsymbol{Z}(I_2))$

Figure 4.1: $\boldsymbol{\mathcal{Z}}-$compatible event trees $\boldsymbol{\mathcal{T}_{\mathcal{Z}}} = \{\mathcal{T}(\boldsymbol{Z}(I_1)), \mathcal{T}(\boldsymbol{Z}(I_2))\}$ yielded by the set of demographic variables $\boldsymbol{\mathcal{Z}} = \{C, V\}$, where $\boldsymbol{Z}(I_1) = (C, V)$ and $\boldsymbol{Z}(I_2) = (V, C)$.

tree $\mathcal{T}(\boldsymbol{Z}(I))$ for some permutation I and its stage partition has the following properties:

1. Each stage only gathers situations that are at the same distance from the root node.

2. For any two situations $s_1(\boldsymbol{z}_1^{(k)})$ and $s_2(\boldsymbol{z}_2^{(k)})$ at the same stage the mapping associating their florets $\mathcal{F}(s_1)$ and $\mathcal{F}(s_2)$ always maps the edges so that their labels $z_{i_{k+1}} \in \mathcal{Z}_{i_{k+1}}$ on the full tree coincide.

The first condition implies that each stage can only gather situations $s(\boldsymbol{z}^{(k)})$, $\boldsymbol{z}^{(k)} \in \mathbb{Z}^{(k)}(I)$, associated with the same variable Z_{k+1}, $k = 1, \ldots, N-1$, in $\boldsymbol{\mathcal{Z}}$. Observe that in a SCEG the root situation s_0 will always constitute the singleton stage $u_0 = \{s_0\}$ and the root position $w_0 = \{s_0\}$. The second condition simply demands that when two situations $s_1(\boldsymbol{z}_1^{(k)})$ and $s_2(\boldsymbol{z}_2^{(k)})$ are at the same stage the conditional probability distributions of variables $X(s_1)$ and $X(s_2)$ associated with their florets are then the same for all $z_{i_{k+1}} \in \mathbb{Z}_{i_{k+1}}$:

$$p(X(s_1) = z_{i_{k+1}}|s_1) = p(X(s_2) = z_{i_{k+1}}|s_2).$$

So the meaning of edges associated with the same variable Z_k cannot be permuted to constitute a stage under this condition. The SCEG class is illustrated in the two examples below.

Example 5 (Train Booking with two demographic variables - cont.). In Example 5 the event tree $\mathcal{T}(\boldsymbol{Z}(I_1))$ supports two $\boldsymbol{\mathcal{Z}}$-SCEGs whilst the event tree $\mathcal{T}(\boldsymbol{Z}(I_2))$ supports five $\boldsymbol{\mathcal{Z}}$-SCEGs. To see this, take first the event tree $\mathcal{T}(\boldsymbol{Z}(I_1))$. From condition 1 in Definition 22 we cannot merge the root situation s_0 with any other situation. In other words, we only merge situations associated with the same variable. Condition 2 in Definition 22 implies that there is only one way to merge situations s_1 and s_2: the probability of a particular event to happen should be the same whether a tourist is at situation s_1 or s_2.

For instance, we are not allowed to gather situations s_1 and s_2 if the probability vector on the edges (w, m, s) of s_1 is identical to the probability vector on the edges (m, s, w) of s_2 because the probability matching between these situations are associated with a permutation of edges. To merge these situations into a single position the probability vector on the edges (w, m, s) of s_1 has to be identical to the probability vector on the edges (w, m, s) of s_2. In light of these conditions we only have the following two possible stage structures:

1. $U_a(I_1) = \{u_0 = \{s_0\}, u_1 = \{s_1\}, u_2 = \{s_2\}\}$, *and*
2. $U_b(I_2) = \{u_0 = \{s_0\}, u_1 = \{s_1, s_2\}\}$.

For analogous reasons, it is straightforward to verify that the event tree $\mathcal{T}(\boldsymbol{Z}(I_2))$ only supports five possible $\boldsymbol{\mathcal{Z}}$-SCEGs whose stage structures are:

1. $U_a(I_1) = \{u_0 = \{s_0\}, u_1 = \{s_1\}, u_2 = \{s_2\}, u_3 = \{s_3\}\}$,
2. $U_b(I_2) = \{u_0 = \{s_0\}, u_1 = \{s_1, s_2\}, u_2 = \{s_3\}\}$,
3. $U_c(I_3) = \{u_0 = \{s_0\}, u_1 = \{s_1, s_3\}, u_2 = \{s_2\}\}$,
4. $U_d(I_3) = \{u_0 = \{s_0\}, u_1 = \{s_1\}, u_2 = \{s_2, s_3\}\}$, *and*
5. $U_e(I_5) = \{u_0 = \{s_0\}, u_1 = \{s_1, s_2, s_3\}\}$.

□

Example 6 (Train Booking). *Return again to the example of booking a train*

described in Section 3.1. Any event tree yielded by the set of demographic variables $\boldsymbol{\mathcal{Z}} = \{C, V, A, T\}$ is a $\boldsymbol{\mathcal{Z}}-$compatible event tree and so the event tree corresponding to the staged depicted in Figure 3.2 is a $\boldsymbol{\mathcal{Z}}-$compatible event tree. If we only merge situations into a single stage associated with the same variable every CEG supported by one of this $\boldsymbol{\mathcal{Z}}-$compatible event tree will be a $\boldsymbol{\mathcal{Z}}-$SCEG. This is exactly the case with the CEG showed in Figure 3.3. □

4.1.2 A Prior over the model space

Eliciting a prior distribution for each model is often a demanding task because the size of the model space is immense even when there is only a single spanning event tree with a moderate number of situations. In practise, it is common to make an objective Bayesian assumption. One popular choice is to assume all models equally likely. Formally, this would then give us that

$$q(\mathbb{C}) = \frac{1}{|\boldsymbol{\mathcal{C}}|}, \forall \mathbb{C} \in \boldsymbol{\mathcal{C}}, \tag{4.4}$$

where $|\mathcal{A}|$ is the total number of elements of a set $\mathcal{A}$. In this situation, the log posterior BF ($lpBF$) between two models $\mathbb{C}_1$ and $\mathbb{C}_2$ reduces to the ratio of their marginal likelihood. Thus the best scored model in the CEG model space corresponds to the MAP CEG.

A uniform prior over the model space is computationally very simple to implement. Since this prior assigns a strictly positive probability for all models, it also has a nice asymptotic property (Bernardo and Smith, 2004, Schervish, 1996). If the model space includes the true model, then the model search algorithm will find it with probability one in the long run. Otherwise, the closest model to the true one in Kullback-Liebler distance will be selected. In the BN contexts, this uniform prior meets the widely accepted requirement that Markov (or likelihood) equivalent models (see Section 2.5) should have the same prior (Heckerman et al., 1995, Heckerman, 1999). However, the conditions to fully justify this prior can be disputable in real-world applications.

An alternative way to satisfy the Markov equivalence condition is to set a uniform

prior over the class of Markov equivalent models. Explicitly, we then have that

$$q(\mathbb{C}) = \frac{1}{|M(\boldsymbol{\mathcal{C}})|}, \forall \mathbb{C} \in M(\boldsymbol{\mathcal{C}}), \tag{4.5}$$

where $M(\boldsymbol{\mathcal{C}}) = M_1, \ldots, M_m$ is a partition of the model space $\boldsymbol{\mathcal{C}}$ such that any two CEGs in the same set M_i are Markov equivalent whilst any two CEGs chosen from different sets M_a and M_b, $a \neq b$, are not Markov equivalent. Nevertheless the use of uniform priors based on Markov equivalent models is not universally supported. Some argue that models can be distinguishable even though they embed the same conditional independence structures (Korb and Nicholson, 2011).

In practise I have found during my analyses using CEG models that such decisions do not appear to be critical to the ensuing inference. Also note that when a unique given event tree defines the explored CEG model space $\boldsymbol{\mathcal{C}}$ (i.e. $|\boldsymbol{\mathcal{T}}| = 1$) no pair of CEGs can be Markov equivalent. Therefore in this case the use of a uniform prior over the model space is not so controversial as it could otherwise be.

Recall that the posterior Bayes factor is a relative approach (Cowell et al., 2007), providing the framework to conduct pairwise model search. If it is possible to execute all possible comparisons over the whole family $\boldsymbol{\mathcal{C}}$, the best scored model will be found. Otherwise, some bespoke approximative algorithm should be designed at least to assure that a good model will be chosen. Comprehensive reviews about BF are given in Kass and Raftery (1995), Berger and Pericchi (2001) and Pericchi (2005). For BF model selection over linear and CART models, expert systems and dynamic models, see Chipman et al. (2001), Cowell et al. (2007) and West and Harrison (1999), respectively.

In our case, comparing two proposal CEGs $\mathbb{C}_1$ and $\mathbb{C}_2$ can be done equivalently by comparing their log posterior Bayes factors with those for a third CEG $\mathbb{C}_3$ as follows

$$\begin{aligned} \log pBF(\mathbb{C}_1, \mathbb{C}_2) &= \log p(\mathbb{C}_1|\boldsymbol{x}) - \log p(\mathbb{C}_3|\boldsymbol{x}) - \log p(\mathbb{C}_2|\boldsymbol{x}) + \log p(\mathbb{C}_3|\boldsymbol{x}) \\ &= \log pBF(\mathbb{C}_1, \mathbb{C}_3) - \log pBF(\mathbb{C}_2, \mathbb{C}_3). \end{aligned} \tag{4.6}$$

This property enables us to design an efficient model search algorithm using a

heuristic strategy which restricts the model search space at each iteration to a local neighbourhood $\boldsymbol{\mathcal{N}}(\mathbb{C}_3)$, $\boldsymbol{\mathcal{N}}(\mathbb{C}_3) \subseteq \boldsymbol{\mathcal{C}}$, around the current best scored CEG $\mathbb{C}_3$. To select the best scored model in $\boldsymbol{\mathcal{N}}(\mathbb{C}_3)$ it is necessary only to compare all models in $\boldsymbol{\mathcal{N}}(\mathbb{C}_3)$ against $\mathbb{C}_3$ regardless of the prior distribution imposed over the model space. This often requires fewer calculations than computing the score associated with all pairs of models in $\boldsymbol{\mathcal{N}}(\mathbb{C}_3)$ especially if all models in $\boldsymbol{\mathcal{N}}(\mathbb{C}_3)$ are nested into $\mathbb{C}_3$ in the sense of Definition 23.

Definition 23 (m-Nested Chain Event Graphs)**.** A CEG $\mathbb{C}^+ = (\mathcal{T}, U^+, \mathscr{P})$ is m-nested in any CEG $\mathbb{C} = (\mathcal{T}, U, \mathscr{P})$ if and only if U is a finer partition of U^+ and $|U| - |U^+| = m$. Conventionally $\Delta(U, U^+)$ is the set of stages of U that are merged in U^+.

4.1.3 A Prior over the parameter space

To set priors over the parameter space of every CEG in the model space can be a challenging task. As discussed in Section 3.3.1 this can be facilitated in the case of a standard CEG if we fix a hyper-parameter $\bar{\alpha}$ and propagated it over the CEG under uniform and conserving conditions. If we plan to search over a collection of CEGs yielded by the same event tree we will have to choose a hyper-parameter $\bar{\alpha}$ for each CEG and a natural option is to fix the same value of this hyper-parameter for all CEGs.

Remarkably Freeman and Smith (2011a) obtained a formal Dirichlet characterisation of CEGs that resemble the Dirchlet characterisation of BNs (Heckerman and Geiger, 1995, Geiger and Heckerman, 1997, Heckerman, 1999). They showed that the product Dirichlet priors and the conserving propagation of the hyper-parameter $\bar{\alpha}$ are inevitably for all CEGs supported by the same event tree if the three conditions below hold.

Path Independence In the CEG $\mathbb{C}_0$ units take the paths from the root vertex to a leaf vertex at independent rates.

Floret Independence The probability of which edge a unit takes immediately

after arriving at a stage u in the CEG $\mathbb{C}_0$ is independent from the rate that units arrive at this stage u.

Margin Equivalence Any pair of equal stages in two distinct CEG models have an identical prior probability distribution: $p(\boldsymbol{\pi}|\mathbb{C}_1) = p(\boldsymbol{\pi}|\mathbb{C}_2)$.

Here the CEG $\mathbb{C}_0$ is the CEG that has the finest stage structure supported by an event tree, i.e. each situation in the event tree constitutes a single stage in $\mathbb{C}_0$. Note that when a CEG model is also a BN model and the above three assumptions are valid, the hyper-parameter $\bar{\alpha}$ is completely analogous to the concept of equivalent sample size in the BN framework (see Section 2.5). Setting the hyper-parameter $\boldsymbol{\alpha}$ for each CEG $\mathbb{C}$ in the model space we then need only to set the hyper-parameter $\boldsymbol{\alpha}^0$ for $\mathbb{C}_0$ since for all $i = 1, \ldots, M$ and $j = 1, \ldots, L_i$ associated with the stage structure of $\mathbb{C}$ we have

$$\alpha_{ij} = \sum_{s_k \in u_i} \alpha^0_{kj}. \tag{4.7}$$

As a *new contribution*, I propose to extend this Dirichlet characterisation for the set $\boldsymbol{\mathcal{C}}$ of $\boldsymbol{\mathcal{Z}}-$SCEGs. For this purpose I need to adopt two additional conditions described below. These assumptions are completely analogous to those adopt for the BN learning (Heckerman and Geiger, 1995, Geiger and Heckerman, 1997, Heckerman, 1999).

Structure Possibility For every $\boldsymbol{\mathcal{Z}}-$compatible event tree its corresponding CEG $\mathbb{C}_0$ has prior probability strictly positive.

Likelihood (or Markov) Equivalence Every pair of CEG $\mathbb{C}_0 \in \boldsymbol{\mathcal{C}}$ should have to have the same marginal likelihood.

It then follows directly from these two conditions and the result in Freeman and Smith (2011a) that any two CEGs $\mathbb{C}_0 \in \boldsymbol{\mathcal{C}}$ will have the same hyper-parameter $\bar{\alpha}$. I have noted that these two additional conditions were implicitly assumed by Cowell and Smith (2014) to search the SCEG model space. On the basis of these results I am able to propose a *new definition* of a standard CEG model selection.

Definition 24 (Standard CEG model selection)**.** The search over the CEG model space using the Bayesian framework where the assumptions of path and floret

independences, margin equivalence, structure possibility, likelihood equivalence and complete random sampling hold is called a *standard CEG model selection*.

In real-world applications, the assumptions for a standard CEG model selection should be careful verified and validated. For example, a complete random sampling can be obtained by an appropriate experimental design and rigorous protocol for data collection. We often justify path and floret independences as well as margin equivalence based on domain knowledge. Heckerman (2008) argued that likelihood equivalence is a reasonable assumption for observational studies - the focus of this thesis. In this case, this assumption only asserts that data does not help us to distinguish two different models that entail the same set of conditional independence structures. Structure possibility is a mild condition which keeps the model space as large as possible in terms of supported event trees.

4.2 Greedy CEG Model Search

In this section I will discuss some approximate model search algorithms based on agglomerative hierarchical clustering (AHC) technique. These explore the CEG model space $\mathcal{C}$ by adopting the standard Bayesian CEG model selection assumption. I will distinguish between two contexts usually found in real-world applications. The first case is when domain experts are able to fully construct the event tree that supports the CEG model. The other situation arises when a family of $\mathcal{Z}-$compatible event trees is implicitly defined by a set of random variables $\mathcal{Z} = \{Z_1, \ldots, Z_N\}$. I will also introduce a *new* useful concept called *hyper-stage* that enables us to design more efficient algorithm.

Briefly the AHC method organises data into a hierarchy of clusters. It starts from the singleton clusters initially defined by the data structure. Using some proximity score and usually adopting some greedy strategy the algorithm then proceeds to merge sequentially the clusters until obtaining only one cluster. The result is the organisation of the data into a hierarchical sequence of nested partitions. This is commonly depicted by a dendrogram or a binary tree.

The final clustering is obtained by cutting the dendrogram at a particular level according to some criterion. Of course, if the interest is into this ultimate clustering result then it is not necessary to construct the full hierarchy of clusters, particularly when the cutting criterion is automatically embedded within the clustering algorithm. This is exactly what happens in the CEG model algorithms discussed below. For more detail about general AHC methods, see e.g. Jain and Dubes (1988), Hansen and Jaumard (1997), Jain et al. (1999), Heard et al. (2006) and Xu and Wunsch (2009).

4.2.1 CEG Model Search over a particular event tree

An AHC Algorithm for CEG model selection

To search over the CEG model space $\boldsymbol{\mathcal{C}}$ for any given event tree, Freeman and Smith (2011a) implemented the agglomerative hierarchical clustering (AHC) algorithm using a Bayesian approach; see Algorithm 2 below. This framework explores the Dirichlet characterisation by adopting a CEG greedy search strategy based on the model score provided by the posterior probability of each model.

Assume that the AHC algorithm found the CEG $\mathbb{C}_i$ as the best model in the end of an iteration i, where the model search starts at the CEG $\mathbb{C}_0$ that has the finest stage structure: each stage gathers only one situation. Define the search neighbourhood at iteration i as a family of models $\boldsymbol{\mathcal{N}}_2(\mathbb{C}_{i-1}) = \{\mathbb{C}_j^+\}$ constituted by all 1-nested CEGs $\mathbb{C}_j^+$ in $\mathbb{C}_{i-1}$.

Setting a uniform prior over the model space $\boldsymbol{\mathcal{C}}$ and assuming the validity of the conditions for standard CEG models, the log posterior BF (lpBF) between the initial model $\mathbb{C}_{i-1}$ and a candidate model $\mathbb{C}_j^+ \in \boldsymbol{\mathcal{N}}_2(\mathbb{C}_{i-1})$ at iteration i is simplified by

$$\begin{aligned} lpBF(\mathbb{C}_{i-1}, \mathbb{C}_j^+) &= a(\boldsymbol{\alpha}_1) - a(\boldsymbol{\alpha}_1^*) - b(\boldsymbol{\alpha}_1) + b(\boldsymbol{\alpha}_1^*) + a(\boldsymbol{\alpha}_2) - a(\boldsymbol{\alpha}_2^*) \\ &\quad - b(\boldsymbol{\alpha}_2) + b(\boldsymbol{\alpha}_2^*) - a(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2) + a(\boldsymbol{\alpha}_1^* + \boldsymbol{\alpha}_2^*) \\ &\quad + b(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2) - b(\boldsymbol{\alpha}_1^* + \boldsymbol{\alpha}_2^*), \end{aligned} \tag{4.8}$$

where:

Algorithm 2: AHC Algorithm

Input: A complete data set $\boldsymbol{D}$, an event tree $\mathcal{T}$ and a parameter $\bar{\alpha}$.

Output: The best scoring CEG found.

Initialise the array U with the stage structure of $\mathbb{C}_0$.
Obtain the conditional frequency tables ($\boldsymbol{y}$) for each stage of $\mathbb{C}_0$ based on $\boldsymbol{D}$ and $\mathbb{C}_0$.
Calculate the hyperparameter $\boldsymbol{\alpha}$ for each stages of $\mathbb{C}_0$ using $\bar{\alpha}$ based on conservative and uniform assumptions.
Initialise an array $score$ with the log posterior probability of $\mathbb{C}_0$.
$stop \leftarrow FALSE$
while *stop=FALSE* **do**
    for *every pair of stages* $\{u_a, u_b\}$ *with the same number of outgoing edges* **do**
        Using Equation 4.8 calculate the $lpBF$ between a stage structure that merges the stages u_a and u_b into the same stage keeping all other stages untouched.
        if *there does not exist any pair* $\{u_a, u_b\}$ **then**
            $stop \leftarrow TRUE$
    if *stop=FALSE* **then**
        Take the pair of stages u_a^* and u_b^* that provides the largest $lpBF$.
        if $lpBF[\{u_a^*, u_b^*\}] > 0$ **then**
            $score \leftarrow score + lpBF[\{u_a^*, u_b^*\}]$
            Update U gathering u_a^* and u_b^* into a single stage $u_{a\oplus b}^*$ and eliminating the stage u_b^*.
        else
            $stop \leftarrow TRUE$
return U, $score$

- u_1 and u_2 are the two stages of $\mathbb{C}_{i-1}$ that are merged to obtain $\mathbb{C}_j^+$;
- α_1 and α_1^* are, respectively, the prior and posterior hyper-parameters associated with stage u_1 in $\mathbb{C}_{i-1}$; and
- α_2 and α_2^* are, respectively, the prior and posterior hyper-parameters associated with stage u_2 in $\mathbb{C}_{i-1}$.

So the MAP CEG in $\boldsymbol{\mathcal{N}}_2(\mathbb{C}_{i-1})$ at iteration i corresponds to a model $\mathbb{C}_i$ such that

$$\mathbb{C}_i = \arg\max_{\mathbb{C}_j^+ \in \boldsymbol{\mathcal{N}}_2(\mathbb{C}_{i-1})} \mathbb{C}_j^+. \tag{4.9}$$

The algorithm stops the search when the $lpBF(\mathbb{C}_{i-1}, \mathbb{C}_i)$ is negative.

An Optimised AHC Algorithm for CEG model selection

I have noted that the AHC algorithm (Algorithm 2), as presented in Freeman and Smith (2011a), does not fully explore structures that may be present in the data. These structures are often domain-specific: for example, only to gather situations into a single stage if they are associated with the same random variable. In this section I will develop a formal framework to close this gap which constitutes an *original material* for this thesis.

To perform an exhaustive search I propose splinting the set of situations associated with an event tree into *hyper-stages* $\mathcal{H} = \{\mathcal{H}_h; h = 1, \ldots, H\}$ according to some criteria that I will discuss next. Note that the hyper-stages do not need to be mutually exclusive. I then demand that any two situations s_a and s_b can be merged into a single stage if and only if there exists a hyper-stage $\mathcal{H}_h$ such that $s_a, s_b \in \mathcal{H}_h$.

An obvious condition that a hyper-stage needs to satisfy is that only situations that have the same number of emanating edges can be merged into a single stage. However, in many applications there is also some domain knowledge available that enables us to further refine this loose definition of hyper-stages $\mathcal{H}$. For example, we often want only to merger situations corresponding to the same variable. Note that a hyper-stage also enables us to encode a priori a set of causal and explanatory hypotheses into our model search algorithm.

I will now discuss through an example how to embed domain information using a hyper-stage structure. I will then proceed to show that this *new concept* enables us to identify a model subspace that only has CEGs which are consistent with a set of domain hypotheses. In doing this it allows us to avoid spending computational resources and time to search unnecessarily the whole CEG model space.

Example 6 (Train Booking - cont.). Recall the demographic model of the train booking example described in Section 3.1. Assume that a domain expert elicits the event tree depicted in Figure 3.2 and asks us to learning a CEG model using a particular data set. Without any more domain information, the hyper-stage structure $\mathcal{H}_a$ would be defined by gathering situations with the same number of outgoing edges. Therefore we would have that $\mathcal{H}_a = \{\mathcal{H}_{a1}, \mathcal{H}_{a2}\}$, where $\mathcal{H}_{a1} = \{s_0, s_3, \ldots, s_{20}\}$ and $\mathcal{H}_{a2} = \{s_1, s_2\}$.

However, when we fed back this to our client he said that it did not make sense to merge situations associated with different variables. Also observe that clustering situations in the same path would demand us some domain knowledge to justify the path independence assumed in our standard CEG model selection framework. The hyper-stage structure would then be given by $\mathcal{H}_b = \{\mathcal{H}_{b1}, \mathcal{H}_{b2}, \mathcal{H}_{b3}, \mathcal{H}_{b4}\}$, where $\mathcal{H}_{b1} = \{s_0\}$, $\mathcal{H}_{b2} = \{s_1, s_2\}$, $\mathcal{H}_{b3} = \{s_3, \ldots, s_8\}$ and $\mathcal{H}_{b4} = \{s_9, \ldots, s_{20}\}$.

Now suppose that in a third meeting our client told us that the variable Age and Train are strongly constrained by the variable Visit since he observed that passengers with a weak and a strong propensities to enjoy cruiser ships are never included within the same marketing strategy category. Technically this implies a probabilistic dominance over the set of situations corresponding to the variables Age and Train yielded by the variable Visit. He also said that he would like to have a clear separation between local and overseas passengers in terms of train options. This then means that situations associated with the variable Train that descend from situation s_1 should never be merged to those that unfold from situation s_2. Note that this condition does not have any impact on hyper-stages associated with variables Visit and Age. Therefore these two additional hypotheses define the hyper-structure $\mathcal{H}_c = \{\mathcal{H}_{c1}, \ldots, \mathcal{H}_{c8}\}$, where $\mathcal{H}_{c1} =$

$\{s_0\}$, $\mathcal{H}_{c2} = \{s_1, s_2\}$, $\mathcal{H}_{c3} = \{s_3, s_4, s_6, s_7\}$, $\mathcal{H}_{c4} = \{s_4, s_5, s_7, s_8\}$, $\mathcal{H}_{c5} = \{s_9, s_{10}, s_{11}, s_{12}\}$, $\mathcal{H}_{c6} = \{s_{11}, s_{12}, s_{13}, s_{14}\}$, $\mathcal{H}_{c7} = \{s_{15}, s_{16}, s_{17}, s_{18}\}$ *and* $\mathcal{H}_{c8} = \{s_{17}, s_{18}, s_{19}, s_{20}\}$.

Note that as we move from the hyper-structure $\mathcal{H}_a$ to $\mathcal{H}_b$ and then to $\mathcal{H}_c$ we are reducing the number of possible CEGs in the search model space by encoding some prior domain information. This enable us to find more compelling models for our clients and also to save computational resources. □

Embedding this structural and problem-based information, in some cases it becomes possible to obtain a hyper-stage structure $\mathcal{H}$ that is actually a partition of the set of situations. In this case, an appropriate choice of the CEG score enables us to optimise substantially the CEG model space search. Here it is required that the adopted score has the additive modularity property with regard to each partition $\mathcal{H}_h$.

To formally present this property, let U_h denote the stage structure associated with a hyper-stage $\mathcal{H}_h \in \mathcal{H}$ in a CEG $\mathbb{C}$, where $\mathcal{H}$ is a partition of the set of situations of the event tree supporting $\mathbb{C}$. It then follows from Equation 4.2 that the log posterior probability of $\mathbb{C}$ may be written down in the decomposable form

$$Q(\mathbb{C}) = \sum_{h=1}^{H} Q_{U_h}(\mathbb{C}) \tag{4.10}$$

where $Q_{U_h}(\mathbb{C})$ is the score corresponding to all stages associated with the hyper-stage $\mathcal{H}_h$. Recall that this hyper-stage score is also additively decomposable over the set of stages $u \in U_h$ as follows

$$Q_{U_h}(\mathbb{C}) = \sum_{u \in U_h} Q_u(\mathbb{C}) = \sum_{u \in U_h} \sum_{j=1}^{L(\mathcal{H}_h)} \log \frac{\Gamma(\alpha^*_{uj})}{\Gamma(\alpha_{uj})} - \sum_{u \in U_h} \log \frac{\Gamma(\bar{\alpha}^*_u)}{\Gamma(\bar{\alpha}_u)} \tag{4.11}$$

where $Q_u(\mathbb{C})$ is the log posterior probability of stage u of $\mathbb{C}$ and $L(\mathcal{H}_h)$ is the number of emanating edges of each situation in $\mathcal{H}_h$.

We can therefore maximise this score by maximising the score of each hyper-stage $\mathcal{H}_h \in \mathcal{H}$ independently in spite of calculating the score of all CEGs in each iteration. This is because of the complementary additive decomposition of the

marginal likelihood of each model $\mathbb{C} \in \boldsymbol{\mathcal{C}}$ given in Equation 4.10. Thus, the algorithm needs only to calculate scores corresponding to a partition $\mathcal{H}_h$ that has been optimised in each iteration. I note in passing that, as pointed out by Silander and Leong (2013), some widely used scores such as Bayesian information criterion (Schwarz (1978)), Akaike information criterion (Akaike, 1973) and Bayes Factor (BF) have this property. This fact was also explored in the BN context; see Ott and Miyano (2003), Koivisto and Sood (2004), Singh and Moore (2005) and Silander and Myllymaki (2006).

These adaptations of the previous AHC algorithm can provide a substantial improvement in computational efficiency of time and memory costs. For instance, take a process that unfolds according to a $\boldsymbol{\mathcal{Z}}-$compatible event tree $\mathcal{T}$. Let M_n be the number of situations associated with the n^{th} variable in $\mathcal{T}$ and L_n be the number of edges emanating from its stages. In the worst case, adopting a hyper-stage structure $\mathcal{H} = \{\mathcal{H}_i; i = 1, \ldots, N\}$ such that each partition corresponds to a particular variable reduces the computational complexity from $O(\frac{M_N^4}{L_{N-1}})$ in the original AHC algorithm to $O(M_N^3)$.

With an additional memory cost, the computational time can be further reduced. Consider that under the current iteration i our improved AHC algorithm optimised the staged structure associated with a partition $\mathcal{H}_h$ whose stages are well-ordered. Also assume that our algorithm has a initialise array $lpBF$ that records the score of every possible CEG 1-nested in the highest scored staged structure obtained at iteration $i-1$.

Without loss of generality, assume that stages u_a^* and u_b^*, $a < b$, are merged into a stage $u_{a\oplus b}^*$ at the end of the current step i, where we set $u_{a\oplus b} \equiv u_a$ and eliminate u_b^* in order to kept the stages well-ordered. Under the margin equivalence hypothesis, to update the scores in the array $lpBF$ it is not necessary to recalculate the scores of CEGs obtained from merging a pair of stages in the updated set of stages $U_h \setminus \{u_a^*\}$ since these scores do not change when we merge the previous stages u_a^* and u_b^*. In fact, only the scores of CEGs yielded by merging the the

Algorithm 3: OAHC Algorithm

Input: A complete data set $\boldsymbol{D}$, an event tree $\mathcal{T}$, a hyper-stage structure $\mathcal{H}$ and a parameter $\bar{\alpha}$.

Output: The best scoring CEG found.

Initialise the array U with the stage structure of $\mathbb{C}_0$ indexed by each hyper-stage $\mathcal{H}_h \in \mathcal{H}$, i.e. $|U| = |\mathcal{H}|$.

Obtain the conditional frequency tables ($\boldsymbol{y}$) for each stage of $\mathbb{C}_0$ based on $\boldsymbol{D}$ and $\mathbb{C}_0$.

Calculate the hyperparameter $\boldsymbol{\alpha}$ for each stages of $\mathbb{C}_0$ using $\bar{\alpha}$ based on conservative and uniform assumptions.

Initialise an array $score$ with the log posterior probability of $\mathbb{C}_0$.

for *every partition* $\mathcal{H}_h \in \mathcal{H}$ **do**

    Initialise a vector $lpBF$: for every pair of stages $\{u_a, u_b\} \subseteq U[h]$ lexicographically ordered, calculate the $lpBF$ using Equation 4.8, where the initial model is $\mathbb{C}_0$ and the candidate model merges $u_a = \{s_a\}$ and $u_b = \{s_b\}$.

    $stop \leftarrow FALSE$

    while *stop=FALSE and* $|U[h]| > 1$ **do**

        Take the pair of stages u_a^* and u_b^* that provides the largest score $max(lpBF)$.

        if $max(lpBF) > 0$ **then**

            $score \leftarrow score + max(lpBF)$

            Update $U[h]$: $u_a^* \leftarrow u_{a\oplus b}^*$, where the new stage $u_{a\oplus b}^*$ merges the previous stages u_a^* and u_b^*; and eliminate the stage u_b^*.

            Update $lpBF$: calculate the scores with respect to the new stage u_a^*; and eliminate the scores associated with stage u_b^*.

        else

            $stop \leftarrow TRUE$

return U, $score$

new stage u_a^* and a stage in the updated set $U_h \setminus \{u_a^*\}$ require computations if scores are kept in memory. We also need to eliminate the scores associated with the gathered stage u_b^*.

This procedure in association with an efficient ordering algorithm can reduce the time complexity to $O(M_N^2 \log(M_N))$. For formal details of hierarchical clustering algorithm, see Aggarwal and Reddy (2014) and Manning et al. (2008). Introducing the improvements discussed above, we are able to write a more efficient algorithm for CEG model search, called the Optimised AHC (OAHC) algorithm. This is presented in Algorithm 3 above. Note that Algorithms 2 and 3 are described adopting a uniform propagation of the hyper-parameter $\bar{\alpha}$ over the CEG $\mathbb{C}_0$ (line 3 of these algorithms). Of course, other objective approaches could be embedded within these algorithms or even the analyst could directly provide these algorithms with the hyper-parameter $\boldsymbol{\alpha}$ for $\mathbb{C}_0$. In this last case line 3 of these algorithms could be suppressed.

4.2.2 SCEG Structure learning without a given variable order

When the CEG model space is defined for a particular family of SCEGs based on a set $\boldsymbol{\mathcal{Z}}$ with N discrete random variables, the lack of a given variable order implies that this space is spanned by an enormous collection of $N!$ event trees $\boldsymbol{\mathcal{T}_{\mathcal{X}}}$. The computational complexity will clearly be $O(N!)$ and so this approach quickly becomes intractable even for moderate sized problems.

One possible way to circumvent this issue is to split the model search into two different steps. The first step defines a variable order. The second looks for the best stage structure for its corresponding event tree. Being nested into the SCEG model space, we can use a BN model search algorithm to discover a good variable order in this restricted class. In Barclay et al. (2013) the authors used an exhaustive search algorithm available in the *R* package *deal* (Boettcher and Dethlefsen, 2003). Based on domain knowledge those authors then chose one of the variable orders corresponding to the Markov equivalent MAP BNs. Of course,

other BN model search algorithms based on exhaustive or heuristic approaches could also have been used. For an efficient dynamic programming algorithm to search the BN model space, see Silander and Myllymaki (2006).

The best score BN model provides us with a variable order $\boldsymbol{Z}(I)$ for some permutation I. This defines the event tree $\mathcal{T}(\boldsymbol{Z}(I))$ and so the set of SCEGs $\boldsymbol{\mathcal{C}}(\boldsymbol{Z}(I))$ specified over $\mathcal{T}(\boldsymbol{Z}(I))$. Now the AHC algorithm (or the OAHC algorithm) can be used to search over $\boldsymbol{\mathcal{C}}(\boldsymbol{Z}(I))$ for further asymmetric context-specific conditional statements that might be presented in the data. So the best scored SCEG will correspond to an embellishment of the best found BN. The algorithm is outlined below.

Algorithm 4: BN model search refined using the AHC Algorithm

Input: A complete data set $\boldsymbol{D}$ and a parameter $\bar{\alpha}$.

Output: The best scoring CEG found.

1 Find the best scored BN.

2 Choose one of the variable orders associated with the best scored BNs.

3 Obtain the event tree $\mathcal{T}$ corresponding to the chose variable order $\boldsymbol{Z}(I)$.

4 Find the best CEG using the Algorithm 2 with inputs $\boldsymbol{D}, \mathcal{T}$ and $\bar{\alpha}$.

Since the objectives in Barclay et al. (2013) were to compare the posterior probabilities between the embellished CEG and the best BN, they fixed the same value for the hyper-parameter $\bar{\alpha}$ to initialise the CEG and BN model search algorithms. However, theses values can be different. Sometimes this may be even desirable since the CEG model space is exponentially greater than the BN model space and so slightly greater values of this hyper-parameter may enable us to obtain more stable results.

4.3 Exhaustive CEG Model Search

In this section the dynamic programming search is discussed for the two different contexts discussed for the greedy model search, namely when the model space is defined by a particular elicited event tree and when the model space is constituted

by SCEGs. Here we follow the Bayesian development presented in Cowell and Smith (2014) which aims at finding the MAP SCEG model. For a dynamic programming algorithm using a non-Bayesian additive modular score, see Silander and Leong (2013).

4.3.1 A CEG Model Search with an elicited event tree

I have noted that earlier algorithms (Silander and Leong, 2013, Cowell and Smith, 2014) were developed for the family of SCEG models where the variable order is known. Using the concept of hyper-stage developed previously I am now able to naturally extend this framework for a generic family of CEGs supported by an elicited event tree. This constitutes an *original contribution* for this thesis and enables us to explore asymmetric CEG model spaces such as the space of models associated with the PC sequence followed by a tourist to book a tourist train (Sections 3.1 and 3.2).

Assume that the hyper-stage structure $\mathcal{H}$ constitutes a partition of the set of situations associated with an event tree $\mathcal{T}$. We can then optimise each hyper-stage sequentially. Optimising the score associated with a hyper-stage $\mathcal{H}_h \in \mathcal{H}$ is then achieved by computing the scores for every possible configuration of stages U_h, and then selecting the partition that provides us with the highest score.

For this purpose it is first necessary to calculate the score corresponding to every possible subset of situations associated with a hyper-stage $\mathcal{H}_h \in \mathcal{H}$. Note that each subset constitutes a viable stage u in some SCEG $\mathbb{C}$ supported by $\mathcal{T}$. This CEG model search is described in Algorithm 5 below. This framework is completely analogous to that developed by Cowell and Smith (2014) and Silander and Leong (2013) when the supporting tree is a $\boldsymbol{\mathcal{Z}}-$compatible event tree and each hyper-stage is defined by the set of situations associated with the same random variable. Observe that these two conditions define a collection of SCEGs.

Algorithm 5: An exhaustive CEG model search given an event tree

Input: A complete data set $\boldsymbol{D}$, an event tree $\mathcal{T}$, a hyper-stage structure $\mathcal{H}$ and a parameter $\bar{\alpha}$.

Output: The best scoring CEG found.

Initialise an empty array U such that $|U| = |\mathcal{H}|$.

Obtain the conditional frequency tables ($\boldsymbol{y}$) for each situation of $\mathcal{T}$ based on $\boldsymbol{D}$.

Calculate the hyperparameter $\boldsymbol{\alpha}$ for each situation of $\mathcal{T}$ using $\bar{\alpha}$ based on the conservative and uniform assumptions.

$score \leftarrow 0$

for *every partition* $\mathcal{H}_h \in \mathcal{H}$ **do**

  Calculate the local score Q_{U_h} of every possible subset of situations in $\mathcal{H}_h$.

  Find the best scored partition U_h^* given by $Q_{U_h^*}$.

  $U[h] \leftarrow U_h^*$

  $score \leftarrow score + Q_{U_h^*}$

return U, $score$

4.3.2 SCEG structure learning by dynamic programming

Take a process described by a set of N random variables $\boldsymbol{\mathcal{Z}} = \{Z_1, \ldots, Z_N\}$. To search over the $\boldsymbol{\mathcal{Z}}-$SCEG model space I will present the dynamic programming algorithm developed by Cowell and Smith (2014). Recall that a $\boldsymbol{\mathcal{Z}}-$SCEG model is characterised by the following properties:

1. Any variable order $I = (i_1, \ldots, i_N)$ is compatible with the representation of the process using an event tree $\mathcal{T}(I)$.
2. For any variable order I, situations are at the same distance d from the root node in $\mathcal{T}(I)$ if and only if they corresponds to the same variable $Z_{i_{d+1}}, i_{d+1} \in I$, or they are all leaf nodes.
3. A stage only merges situations associated with the same variable.
4. Any subset of situations associated with the same variable can constitute a

stage.

Let $\mathcal{H}(I) = \{\mathcal{H}_1(I), \ldots, \mathcal{H}_N(I)\}$ be the hyper-stage structure when a variable order I is adopted. The definition of a SCEG guarantees that the hyper-stage structure $\mathcal{H}(I)$ is a partition of the set of situations in $\mathcal{T}(I)$ such that each set $\mathcal{H}_j(I), j = 1, \ldots, N$, gathers all situations in $\mathcal{T}(I)$ associated with a variable Z_{i_j} and so only them.

The additive modularity of the log posterior probability of a SCEG then guarantees that removing the last variable $Z_{i_N^*}, i_N^* \in I^*$, from the variable set does not change the actual best variable order I^* for the remaining $N-1$ variables. Explicitly, if $I^* = (i_1^*, \ldots, i_N^*)$ is the best variable order for a CEG model to represent a process described by the variable set $\boldsymbol{\mathcal{Z}}$ then $I_{N-1}^* = (i_1^*, \ldots, i_{N-1}^*), I_{N-1}^* \subset I^*$, is the best variable order for a CEG model to express the subprocess corresponding to the variable set $\boldsymbol{\mathcal{Z}} \backslash \{Z_{i_N^*}\}$.

Example 7 (Train Booking with three demographic variables). *Return to the example of the train booking described in Section 3.1. Suppose that the decision maker wants to understand the interactions between only the demographic variables Country (C), Visit (V) and Age (A). So, the set* $\boldsymbol{\mathcal{Z}} = \{A, C, V\}$ *spans six* $\boldsymbol{\mathcal{Z}}-$*compatible Event Trees given by the six possible different permutations of these variables.*

From equation 4.10 we have that

$$Q(\mathbb{C}) = \sum_{n=1}^{3} Q_{U_n}(\mathbb{C}), \text{ for all } \mathbb{C} \in \boldsymbol{\mathcal{C}}, \tag{4.12}$$

where U_1, U_2 *and* U_3 *are, respectively, the stage structures associated with variables* A, C *and* V. *The score* $Q_{U_i}(\mathbb{C}), i = 1, 2, 3$, *depends on the variable order. However each highest scored stage structure* $U_n, n = 1, 2, 3$, *can be found independently from each other given a known variable order. In particular, if the variable order* $\boldsymbol{Z}(I^*) = (C, V, A)$ *provides the MAP SCEG* $\mathbb{C}^*$, *then we have necessarily that the best variable order for the set* $\boldsymbol{\mathcal{Z}}^2 = \{C, V\}$ *has to be* $\boldsymbol{Z}^2(I^*) = (C, V)$. □

This leads us to a recursive dynamic programming framework where the problem

of finding the best variable order for $N-1$ variables constitutes a subproblem of discovering the best variable order for N variables. Therefore, for every subset $\boldsymbol{\mathcal{Z}}^k = \{Z_{i_1}, \ldots, Z_{i_k}\} \subset \boldsymbol{\mathcal{Z}}, k = 1, \ldots, N$, we need to find the best sink variable $Z_i \in \boldsymbol{\mathcal{Z}}^k$ given that we have already found the best variable order for every subset $\boldsymbol{\mathcal{Z}}^{k-1} \subset \boldsymbol{\mathcal{Z}}^k$. Embedding this recursive structure into a dynamic programming algorithm enables us to search efficiently the whole SCEG model space.

The general algorithm for learning SCEGs is given in Algorithm 6. I will further explain each of its three steps below.

Algorithm 6: Find the best scoring SCEG when no variable order is specified

Input: A complete data set $\boldsymbol{D}$ on a set of N finite discrete variables $\boldsymbol{\mathcal{Z}}$ and a parameter $\bar{\alpha}$.

Output: The best scoring SCEG found.

1 Discover the best sink variable for all 2^N non-empty subsets of $\boldsymbol{\mathcal{Z}}$.

2 Find the best variable order $I^* = (i_1, \ldots, i_N)$.

3 Recover the highest scoring SCEG using I^*.

Step 1: Discover the best sink variable

The Algorithm 7 is the most computationally intensive step of the general dynamic programming algorithm for $\boldsymbol{\mathcal{Z}}-$SCEG model search. It begins by initialising two 2^N-size arrays $scores$ and $sinks$ where each element corresponds to a subset of $\boldsymbol{\mathcal{Z}}$. It then proceeds to determine the best sink variable of each non-empty subset of $\boldsymbol{\mathcal{Z}}$ by examining them in order of increasing size, starting with singleton subsets.

For every variable Z_n in a set $\boldsymbol{\mathcal{Z}}^{k+1}$ it is necessary to calculate the local score of the best staged tree spanned by the set of variables $\boldsymbol{\mathcal{Z}}^k \cup \{Z_n\}$ such that Z_n is the sink variable and $\boldsymbol{\mathcal{Z}}^k = \boldsymbol{\mathcal{Z}}^{k+1} \setminus \{Z_n\}$. To do this, the algorithm first requires a local auxiliary variable $scoreL$ and a function $BLS(Z_n, \boldsymbol{\mathcal{Z}}^k)$. The best local score associated with $\boldsymbol{\mathcal{Z}}^k$ has already been computed and store in $score$ since the algorithm looks at subsets ordered by increasing size. So the function $BLS(Z_n, \boldsymbol{\mathcal{Z}}^k)$ only needs to calculate the score Q_{U_n} of the best stage partition U_n associated

with the sink variable Z_n. Observe that this does not require the best variable order of $\boldsymbol{\mathcal{Z}}^k$.

Algorithm 7: Find the best sink variables for every non-empty subset of $\boldsymbol{\mathcal{Z}}$.

Input: A complete data set $\boldsymbol{D}$ on a set of N finite discrete variables $\boldsymbol{\mathcal{Z}}$ and a parameter $\bar{\alpha}$.

Output: A set-indexed array $sinks$ that for each subset $\boldsymbol{\mathcal{Z}}^k \subset \boldsymbol{\mathcal{Z}}$ returns the sink variable for the highest scoring CEG spanned by $\boldsymbol{\mathcal{Z}}^k$.

```
for k in 1 → n do
    for Z^k ⊂ Z such that |Z^k| = k do
        scores[Z^k] ← 0
        sinks[Z^k] ← −1
        for Z_i ∈ Z^k do
            Z^{k−1} ← Z^k \ {Z_i}
            scoreL ← BLS(Z_i, Z^{k−1}) + scores[Z^{k−1}]
            if sinks[Z^k] = −1 or scoreL > scores[Z^k] then
                scores[Z^k] ← scoreL
                sinks[Z^k] ← Z_i
return sinks
```

Example 7 (Train Booking with three demographic variables - cont.). Return to Example 7. Now suppose that the decision maker asks his analyst to model that problem. The analyst has then decided to present the MAP SCEG $\mathbb{C}^*$ to the decision maker. For this purpose, he uses the dynamic programming algorithm 6. In the first step the algorithm needs to find a triple $(\boldsymbol{\mathcal{Z}}^k, Z_*, q)$ for every set $\boldsymbol{\mathcal{Z}}^k \subseteq \boldsymbol{\mathcal{Z}}$, where Z_* is the best sink variable for $\boldsymbol{\mathcal{Z}}^k$ and q is the highest score associated with $\boldsymbol{\mathcal{Z}}^k$. The algorithm starts from singleton subsets and so it obtains the following triples: $(\{A\}, A, q_1)$, $(\{V\}, V, q_2)$ and $(\{C\}, C, q_3)$.

Next the algorithm examines the sets of size two. For instance, take the set $\boldsymbol{\mathcal{Z}}^2 = \{A, V\}$. To find the best sink variable for this subset it is necessary to compare the highest scored stage tree associated with the variable order given by $\boldsymbol{\mathcal{Z}}^2(I_1) = (A, V)$ against that one yielded by the variable order $\boldsymbol{\mathcal{Z}}^2(I_2) = (V, A)$.

Note that for the variable order $\boldsymbol{Z}^2(I_1)$ we need to compute only the best score q_a associated with V since the score of A has already been computed previously and it is equal to q_1. Analogously for the variable order $\boldsymbol{Z}^2(I_2)$ we have to find only the best score q_b for A. Now assume that $q_1 + q_a < q_2 + q_b$. Then with regard to the set $\boldsymbol{Z}^2$ the best sink variable is A and its highest score is $q_4 = q_2 + q_b$. Doing similar computations for the other two subsets of size two the algorithm provides the following triples: $(\{A, V\}, A, q_4)$, $(\{A, C\}, C, q_5)$ and $(\{V, C\}, V, q_6)$.

To finalise step 1 the algorithm needs to search for the best sink variable in the set $\boldsymbol{Z}$. Of course, there are three possible candidates: A, V and C. For example, take the variable A. In this case, we have to find the score of the MAP staged tree $\mathcal{T}(\boldsymbol{Z})$ when A is the sink variable. This corresponds only to computing the score q_c of A and then adding it to the score q_6 that was previously calculated. This is because the best variable order of a staged tree does not change if the sink variable is eliminated. Repeating the same procedure for the other two candidate variables we can obtain the score q_d for V and q_e for C. Now assume that $q_6 + q_c > q_5 + q_d > q_4 + q_e$. It then follows that the algorithm stores the triple $(\{A, V, C\}, A, q_7)$, where $q_7 = q_6 + q_c$. □

Step 2: Find the best order of the best sinks

Now the algorithm 8 finds the best order of the best sink variables starting with the complete set $\boldsymbol{Z}$. Assume that at iteration $k, k = N, \ldots, 1$, we have to determine the best sink variables for a set of variables $\boldsymbol{Z}^k_{left} \subseteq \boldsymbol{Z}$. For this purpose, the algorithm first recovers the best sink variable Z_{i_k} of $\boldsymbol{Z}^k_{left}$ from the indexed array $sinks$. Next it removes Z_{i_k} from $\boldsymbol{Z}^k_{left}$ and then begins the iteration $k-1$. The variable Z_{i_k} is stored in the k^{th} element of an n dimensional integer indexed array $order$ of variables.

By carrying on these algorithmic iterations in decreasing order, it then follows that by the end of the algorithm the array $order$ contains the variable order for the highest scoring SCEG. It is also straightforward to see that the root variable Z_{i_1}

Algorithm 8: Find the best variable order

Input: The set indexed array $sinks$.

Output: A integer-indexed array of the variable ordering for the highest scoring CEG.

```
1 Z^k_left = Z
2 for i ← n to 1 do
3     order[i] ← sinks[Z^k_left]
4     Z^k_left ← Z^k_left \ {order[i]}
5 return order
```

corresponds to the variable $order[1]$ whilst the terminate variable Z_{i_n} is stored in $order[n]$. The computational complexity of this step is linear in n.

Example 7 (Train Booking with three demographic variables - cont.). In the second step the algorithm needs to find the best variable order. Using the triple calculated in the previous step, this task is now very simple. Starting with the full set $\boldsymbol{\mathcal{Z}}$ it follows that the best sink variable is A. Next the algorithm identifies the best sink variable for the set $\boldsymbol{\mathcal{Z}}^2_{left} = \boldsymbol{\mathcal{Z}} \backslash \{A\}$. So the result is V since $\boldsymbol{\mathcal{Z}}^2_{left} = \{C, V\}$. Finally the updated set $\boldsymbol{\mathcal{Z}}^1_{left} = \boldsymbol{\mathcal{Z}}^2_{left} \backslash \{V\} = \{C\}$ is a singleton set and so this step terminates. The best variable order is then given by $\boldsymbol{Z}(I^*) = (C, V, A)$. □

Step 3: Recover the highest scoring SCEG

Having found the best variable order I^*, we have first to define its corresponding event tree $\mathcal{T}(\boldsymbol{Z}(I^*))$ and hyper-stage structure $\mathcal{H}$. Next it is necessary only to apply Algorithm 5 to recover the highest scoring SCEG. See Algorithm 9 below.

Algorithm 9: Recover the highest scoring SCEG

Input: A complete data set $\boldsymbol{D}$ on a set of N finite discrete variables $\boldsymbol{\mathcal{Z}}$, a parameter $\bar{\alpha}$ and the best variable order $I^* = (i_1, \ldots, i_N)$.

Output: The best scoring SCEG found.

1 Define the event tree $\mathcal{T}(\boldsymbol{Z}(I^*))$.

2 Define the hyper-stage structure $\mathcal{H}$ for $\mathcal{T}(\boldsymbol{Z}(I^*))$, where each set $\mathcal{H}_n$ corresponds to the variable Z_{i_n}.

3 Obtain the best SCEG using the Algorithm 5 with inputs $\boldsymbol{D}$, $\mathcal{T}(\boldsymbol{Z}(I^*))$, $\mathcal{H}$ and $\bar{\alpha}$.

This dynamic programming algorithm for CEG model search (Cowell and Smith, 2014) closely resembles the dynamic programming algorithm for BN learning (Silander and Myllymaki, 2006). The main difference is that in the dynamic programming algorithm for BN model selection there is a pre-processing step where all local scores are pre-computed. Therefore the MAP BN can be recovered quite directly and at little extra cost. This is because we do not need to run an algorithm given the best BN variable order to find the best parent configuration for each variable: the parent set associated with each variable is actually stored in memory by the algorithm. So instead of recalculating this quantity, the dynamic programming algorithm for CEG learning calculates the local scores as required and caches them. Despite the additional computational time required by the third step to recover the best CEG, the three-step approach adopted for CEG model search is justified because of its much reduced memory cost and also its computational simplicity.

The SCEG model space is far larger than the BN model space. So there are many more partitions in it whose scores should need to be computed and stored. In the BN framework a local score for a variable Z_k is calculated based on an unordered set of its parents. This implies that given a variable Z_k and a subset of variables $\boldsymbol{\mathcal{Z}}^k$ storing the best set of parents for Z_k in $\boldsymbol{\mathcal{Z}}^k$ together with the best local store is computationally cheap. The same observation does not hold for CEGs because whilst the variable order of $\boldsymbol{\mathcal{Z}}^k$ does not change the score Q_{U_k} associated with the sink variable Z_k it does alter its stage structure U_k. In fact, different variable

orders permute the leaf nodes of the event tree spanned by $\boldsymbol{\mathcal{Z}}^k \cup \{Z_k\}$, where Z_k is the last variable. Therefore, a fast recovery of the MAP SCEG would require us to store the best stage configuration for every pair $(Z_k, \boldsymbol{\mathcal{Z}}^k)$, where $\boldsymbol{\mathcal{Z}}^k$ is a possible ordered sequence of $\boldsymbol{\mathcal{Z}}^k$. This would add a complexity of factorial order in the algorithm and so can often exceed the cost of running the Algorithm 5 to recover the MAP SCEG.

4.4 Challenges and Technical Advances for CEG Model Selection

Learning a BN corresponds to learning a restricted set of partitions which prevents us to explore the context-specific conditional independences and possible asymmetries in the development of a process. In contrast, the CEG model space is structurally more flexible. It is therefore more expressive in terms of graphical representation of conditional independences. However, these advantages come at a computational cost for CEG model selection because a CEG probability space with a moderate number of atoms is absolutely gigantic and dwarfs its corresponding BN model space by orders of magnitude.

Thus consider a $\boldsymbol{\mathcal{Z}}$-SCEG model space $\boldsymbol{\mathcal{C}}$ spanned by a set of $N, N \geq 2$, discrete random variables $\boldsymbol{\mathcal{Z}} = \{Z_1, Z_2, \ldots, Z_N\}$, where each random variable Z_n has L_n finite number of categories. Let $M_{n(I)}$ be the number of situations associated with the n^{th} variable X_{i_n} in an $\boldsymbol{\mathcal{Z}}$-compatible event tree $\mathcal{T}(\boldsymbol{X}(I))$. Since in $\mathcal{T}(\boldsymbol{X}(I))$ every situation at distance k from the root situation s_0 has L_{k+1} children, it then follows that $M_{n(I)} = 1$, if $n = 1$, and $M_{n(I)} = \prod_{j=1}^{n-1} L_{i_j}$, if $n = 2, \ldots, N$. The total number of partitions of these situations is then given by the $M^{th}_{n(I)}$ Bell number $B_{M_{n(I)}} = \sum_{i=0}^{M_{n(I)-1}} \binom{M_{n(I)-1}}{i} B_i$ (Spivey, 2008). Recall that the Bell numbers B_i grows very fast with i; for instance, $B_1 = 1$, $B_2 = 2$, $B_4 = 15$, $B_8 = 4,140$ and $B_{16} \approx 10^{10}$. Now remember that each partition constitutes a different stage structure U_n and so a distinct SCEG model corresponding to a variable Z_{i_n}. Considering the $N!$ variable orders it then follows that the size of

the SCEG model space is written down by

$$|\mathcal{C}| = \sum_{I \in \mathcal{I}} \prod_{n=1}^{N} B_{M_{n(I)}}, \tag{4.13}$$

where $\mathcal{I}$ is the set of all possible permutations I.

This implies that the complexity of this space grows exponentially in terms of Bell numbers and depends on not only the number of variables but also the number of categories that each variable has. Therefore, searching over the SCEG model space is enormously more challenging computationally than searching over its corresponding BN model space: there are far more partitions to explore.

For instance, consider a process defined by a set of four binary random variables whose order is known. Learning a BN model requires us to calculate only 15 ($\sum_{i=1}^{4} 2^{i-1}$) local scores whilst learning a SCEG model implies the computation of 4,158 ($\sum_{i=1}^{4}\{B_{2^{i-1}} - 1\}$) local scores. Note that in this simple example learning a SCEG model demands the calculation of 277 times more local scores than learning a BN model and so it requires much more computational time and memory resource. Of course, the computation of CEG local scores can be abbreviated if we use the fact that these 4,158 scores are yielded by only 279 distinct sub-partition scores. Even in this case, we must compute 19 times more scores for learning a CEG model than for learning a BN model. On the other hand, this approach implies to spend more memory resource since we have to store the sub-partition scores. However, this extra memory cost is more than justified by computational time saving. For empirical studies about computational time required to learn a CEG model, see Silander and Leong (2013). Further discussion about computational cost associated with learning a CEG model can also be found in Cowell and Smith (2014).

Therefore, the dynamic programming search method quickly becomes infeasible as the number of random variables in $\boldsymbol{\mathcal{Z}}$ increases to an even moderate size. In this case, heuristic search strategies such as the agglomerative clustering are needed to scale up the size of the SCEG model space to search over (Silander and Leong, 2013, Cowell and Smith, 2014). A promising fast approximation is

to embed the heuristic within the dynamic programming algorithm (Silander and Leong, 2013). Exploring this alternative, Silander and Leong (2013) were able to search over model space defined by up to 18 random variables in less than 10 minutes. Those authors showed empirically that the AHC approach performed better than K-mean clustering methods when they are used in conjunction with the dynamic programming model search. However the AHC algorithm is much slower.

To implement these approximations, I have note that it is necessary only to rewrite the function $BLS(Z_i, \boldsymbol{\mathcal{Z}}^k)$ used in Algorithm 7. Instead of looking at the scores of all possible stage structures this function will now find the best stage partition U_n associated with the variable X_n in a set $\boldsymbol{\mathcal{Z}}^k$ using the adopted heuristic algorithm.

During the modelling process the identification of a partial order for the variables in $\boldsymbol{\mathcal{Z}}$ based on the domain information may enable modellers to reduce the computational complexities in these full search methods. Particularly, the definition of a block order as I propose in Definition 25 provides us with a well-ordered partition of $\boldsymbol{\mathcal{Z}}$. This enables us to greatly reduce the space of allowed models that the search needs to be carried out on. Thus, to find the highest scoring SCEG it suffices to look over the CEG model subspace constituted by those obtained by permuting the variables within each block $\boldsymbol{\mathcal{B}}_b$, $b = 1, \ldots, B$.

Definition 25 (Variable Block Order)**.** Take a set $\boldsymbol{\mathcal{Z}} = \{Z_1, \ldots, Z_n\}$ of N discrete random variables. A *block order* of $\boldsymbol{\mathcal{Z}}$ is a partition $\boldsymbol{\mathcal{B}} = (\boldsymbol{\mathcal{B}}_1, \ldots, \boldsymbol{\mathcal{B}}_B)$, such that a $\boldsymbol{\mathcal{Z}}-$SCEG $\mathbb{C}(\mathcal{T}(I))$, where $I = (i_1, \ldots, i_N)$, has non-zero probability a priori if and only if for any pair of variables $Z_{i_n} \in \boldsymbol{\mathcal{B}}_{b_1}$ and $Z_{i_{n+1}} \in \boldsymbol{\mathcal{B}}_{b_2}$, $n = 1, \ldots, N-1$, we have that $b_1 \leq b_2$.

The Algorithm 10 that I developed implements this idea by adding a loop in the Algorithm 7 to control for the blocks. Note that the function BLS and the other steps of the algorithm do not change. Also observe that the concept of block order and its corresponding algorithmic implementation constitute new developments for this thesis.

Algorithm 10: Find the best sink variables for every non-empty subset of $\boldsymbol{\mathcal{Z}}$ consistent with a block ordering $\boldsymbol{\mathcal{B}}$.

Input: A complete data set $\boldsymbol{D}$ on a set of N finite discrete variables $\boldsymbol{\mathcal{Z}}$, a block ordering $\boldsymbol{\mathcal{B}} = (\boldsymbol{\mathcal{B}}_1, \ldots, \boldsymbol{\mathcal{B}}_B)$ and a parameter $\bar{\alpha}$.

Output: A set-indexed array $sinks$ that for each subset $\boldsymbol{\mathcal{Z}}^l \subset \boldsymbol{\mathcal{Z}}$ consistent with the block ordering returns the sink variable for the highest scoring SCEG spanned by $\boldsymbol{\mathcal{Z}}^l$.

```
1  l ← 0
2  for b in 1 → B do
3      for k in 1 → |B_b| do
4          l ← l + 1
5          for B_b^k ⊂ B_b such that |B_b^k| = k do
6              Z^l = ⋃_{j=0}^{b-1} B_j ∪ B_b^k, where B_0 = ∅
7              scores[Z^l] ← 0
8              sinks[Z^l] ← −1
9              for Z_i ∈ B_b^k do
10                 Z^{l(−1)} ← Z^l \ {Z_i}
11                 scoreL ← BLS(Z_i, Z^{l(−1)}) + scores[Z^{l(−1)}]
12                 if sinks[Z^l] = −1 or scoreL > scores[Z^l] then
13                     scores[Z^l] ← scoreL
14                     sinks[Z^l] ← Z_i
15 return sinks
```

Parallel computation is a good option that can speed up exhaustive model searches. I now briefly propose some *original ways* to implement this using the algorithms discussed previously. The key observation here is that the local scores Q_{U_n} associated with a variable Z_{i_n} at level ℓ_{n-1} in the event tree can be independently computed from the local scores of variables at other levels. The speed-up gain can be substantial especially for the last levels of large event trees. When a variable order is known, the loop over the sequence of variables $\boldsymbol{Z}(I)$ (line 5 of Algorithm 5) can be directly parallelised. In the case of a full search without a variable order

the parallel programming can be easily implemented over the intra-level loop to find the best sink variables. This corresponds to parallelising the computation of the inner loop over the set of variables $\boldsymbol{\mathcal{Z}}_n$ (line 5 of Algorithm 7). If we have a block order, parallel computation can then be introduced over the blocks (line 2 of Algorithm 10) and inside the blocks (line 7 of Algorithm 10).

4.5 Some Computational Experiments

In this section, I will find the MAP CEG models associated with the train booking example described in Section 3.1 using the exhaustive model search algorithms presented in Section 4.3. I will then explain how to read and interpret the conditional independence hypotheses embedded within these models. This is the *first study where* this train booking process is analysed.

To initialise the search algorithms I first assessed values of the hyper-parameter $\bar{\alpha}$ in the range $\{1, 2, \ldots, 15\}$. I also assumed the uniform condition to propagate this hyper-parameter over the event tree. Finally, this hyper-parameter was empirically fixed at 3 for both models because this value corresponds to the smallest hyper-parameter $\bar{\alpha}$ that enables us to obtain stable results; i.e, the results are not sensitive to values of this hyper-parameter greater than 2. A small value of the hyper-parameter $\bar{\alpha}$ also allows us to enforce a plausibly large variance over the prior marginal distribution of each variable. This is important since we do not have any domain knowledge to help us to fix it. Previous CEG (Barclay et al., 2013, Collazo and Smith, 2016) and Bayesian Network (Neapolitan, 2004) studies corroborate with this choice. For a more detail discussion about how to set this hyper-parameter based on empirical studies, see Section 5.4.

4.5.1 PC Sequence Model

The first model represents the different search PC paths that a tourist can choose to follow in order to book a scenic train trip. Being based on a tree, the CEG framework has the necessary flexibility to represent directly the asymmetric un-

foldings that characterises this process. Observe that there are many events with probability zero in its event tree; see the dashed shape nodes in Figure 3.1. Omitting the corresponding edges allows us to further simplify the graphical topology of our model and also the computational complexity of the model search.

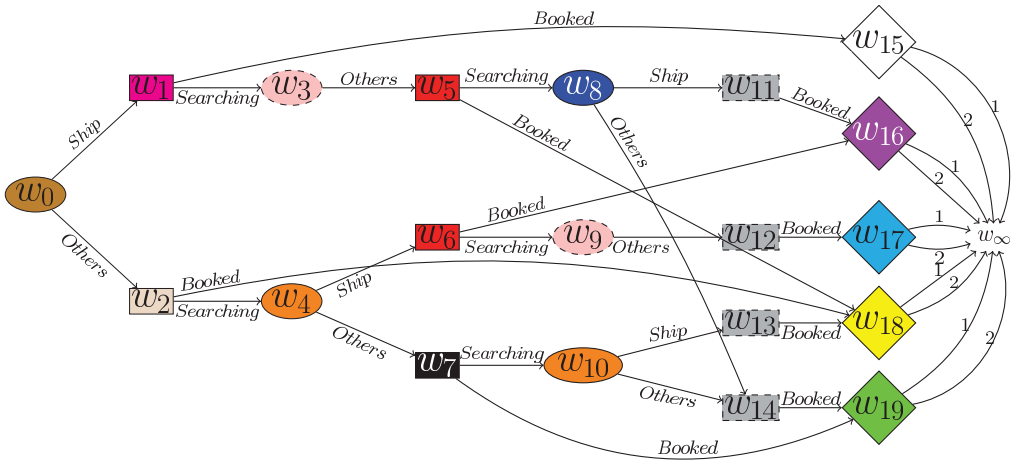


Figure 4.2: The best scored CEG for the PC sequence model. The stage partition is given by: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3, w_9\}$, $u_4 = \{w_4, w_{10}\}$, $u_5 = \{w_5, w_6\}$, $u_6 = \{w_7\}$, $u_7 = \{w_8\}$, $u_8 = \{w_{11}, w_{12}, w_{13}, w_{14}\}$, $u_9 = \{w_{16}\}$, $u_{10} = \{w_{15}, w_{17}\}$, $u_{11} = \{w_{18}\}$, $u_{12} = \{w_{19}\}$. No tourists went through position w_{15}.

Having a well-defined event tree a priori and assuming a hyper-stage for each set of situations that have the same geometric shape in Figure 3.1, I use the Algorithm 5 to look for the best stage structure configuration. Figure 4.2 depicts the MAP CEG model and Table 4.1 shows the posterior conditional probability mean of each stage with a 95% credible interval. The conditional probabilities of stages u_3 and u_8 are degenerated due to the domain conditions. Although the situations s_3 and s_{23} are in the same stage u_{10} and so in the same position, I depict them using two different positions w_{15} and w_{17} to highlight the fact no tourist visited situation s_3 in Figure 3.1 and so the position w_{15} in Figure 4.2.

When a client starts examining his option to book a train ticket (position w_0), it is equally likely that he decides to visit a PC Ship or a PC Others. However this initial choice has a strong impact on the tourist's choice of which PC to book

a train. To understand how we can read this from our CEG model, consider a client that initially visits a PC Others (position w_2). Despite it being possible to book a train at this point this is not likely to happen. This is because there is a clear predisposition (93%) to visit another PC. Note that stage u_4 clearly favours a PC Others (70%). It also plays a key role in this branch of the CEG model since it contains positions w_4 and w_{10}. Finally, observe that a tourist at position w_6 (stage u_5) has a small probability (25%) of booking a train. So we can conclude that if a tourist goes first to a PC Others he will then probably book a train in a PC Others.

Stage	**State Space**	**Mean (95% credible interval)** (%)			
u_0	(Ship,Others)	52	(47,57)	48	(43,53)
u_1	(Booked,Searching)	0.4	(0,2)	99.6	(99,100)
u_2	(Booked,Searching)	7	(4,11)	93	(89,96)
u_3	(Ship,Others)	0	(0,0)	100	(100,100)
u_4	(Ship,Others)	30	(25,36)	70	(64,76)
u_5	(Booked,Searching)	25	(20,30)	75	(70,80)
u_6	(Booked,Searching)	56	(47,64)	44	(36,53)
u_7	(Ship,Others)	83	(71,92)	17	(8,29)
u_8	(Booked,Searching)	0	(0,0)	100	(100,100)
u_9	(1,2)	14	(8,23)	86	(77,92)
u_{10}	(1,2)	62	(38,83)	38	(17,62)
u_{11}	(1,2)	35	(28,42)	65	(58,72)
u_{12}	(1,2)	94	(88,97)	6	(3,12)

Table 4.1: Posterior mean and 95% credite intervals for the stages corresponding to the best scored PC sequence CEG depicted in Figure 4.2.

On the other hand, there is a good chance ($62\% \equiv 0.996 \times 0.75 \times 0.83$) that a tourist who initially visits a PC Ship (position w_1) actually books a scenic train in this PC. However this is only likely to happen after visiting a PC Others. Being at

position w_1 there is a very tiny probability (0.4%) that he will book a train there. So he probably proceeds to position w_5 and visits a PC Others (position w_5) where he has a strong inclination (75%) to carry on searching (position w_8) and so a great tendency (83%) to return to the PC $Ship$.

Recall that there are an extremely larger numbers of PCs gathered in the category Others compared with the category Ship. Therefore we can hypothesise that visiting a PC Others does not influence the subsequent tourist's choice between going to a PC Others or a PC Ship, if it is the case. This allows us to justify the fact that the MAP CEG model gathers positions w_4 and w_{10} at the same stage. So apparently the variables $PC_i, i = 2, 3$ associated with the decisions of which PC to go given that the previous visited PC was a non-cruise PC ($PC_{i-1} = Others$) have indistinguishable conditional probability distributions.

Note that since positions w_5 ($PC_1 = s, PC_2 = o$) and w_6 ($PC_1 = o, PC_2 = s$) are at the same stage the order of the two first visited PCs does not have any impact on the second decision with respect to keep searching or to book a train. Also observe that 90% of train bookings during the second visit happens in a PC Others.

Lastly, the MAP CEG model suggests that there are four different groups of tourists (positions $w_{16}, \ldots, w_{19}$) with respect to the option between public or cruise trains. The great majority of clients (94%) who visit two non-cruise PCs successively before booking a train consecutively (position w_{19}) prefer public trains. Having visited a PC Others, a PC Ship and a PC Others consecutively (position w_{17}) the chance that a client books a public train is reduced to 62%.

In contrast, the other two groups present a strong preference for cruise trains. Most tourists (65%) booking their trains in a PC Others without visiting two non-cruise PCs - sequences (o), (s,o) at position w_{18}- tend to book a cruise train. This also happens with clients who arrive in a PC Ship after visiting two non-cruise PCs successively -sequence (0,0,s) at position w_{18}. Having not previously gone to two PCs Others passengers in a PC Ship (position w_{16}) -sequences (s,o,s) and (o,s)-

have the highest probability (86%) to book a cruise train.

4.5.2 Demographic Model

Remember that we can obtain six $\mathcal{Z}-$compatible event trees, where the set of variables is given by $\mathcal{Z} = \{C, V, A, T\}$. However, we do not need to search over the whole CEG model space because the last variable is fixed at T based on domain knowledge. So, there is a block ordering $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2\}$, where $\mathcal{B}_1 = \{C, V, A\}$ and $\mathcal{B}_2 = \{T\}$. To find the variable order that provides us with the MAP CEG, we then search the CEG model space using the Algorithm 10.

The variable orders $I_1 = (C, V, A, T)$ and $I_2 = (V, C, A, T)$ enables us to construct equally best scored models that are also statistically equivalent. We believe that the order I_1 is more appropriate for this particular example for two reasons.

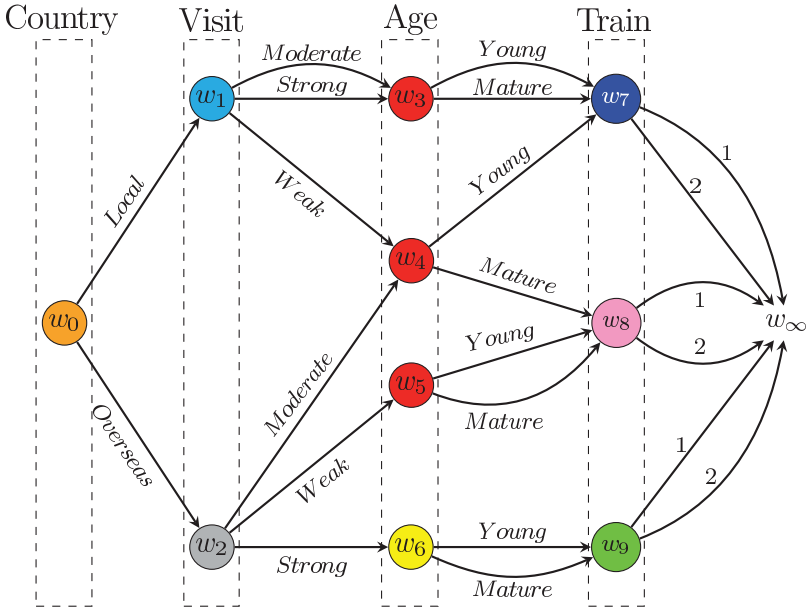


Figure 4.3: The MAP SCEG corresponding to the train booking process when the demographic variables are taken into consideration. Variable order $I_1 = (C, V, A, T)$. The stage structure is given by: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3, w_4, w_5\}$, $u_4 = \{w_6\}$, $u_5 = \{w_7\}$, $u_6 = \{w_8\}$, $u_7 = \{w_9\}$. This CEG is identical to one depicted in Figure 3.3.

First, its corresponding MAP CEG graph is much simpler in terms of number of nodes and edges than the MAP CEG associated with the order I_2. This hap-

pen because the variable V has three categories whilst variable C only has two categories. Therefore, the $\boldsymbol{\mathcal{Z}}-$compatible event tree $\mathcal{T}(\boldsymbol{\mathcal{Z}}(I_1))$ is topologically much simpler with respect to their second level than the $\boldsymbol{\mathcal{Z}}-$compatible event tree $\mathcal{T}(\boldsymbol{\mathcal{Z}}(I_2))$.This graphical simplification is naturally reflected into the MAP CEG $\mathbb{C}(\boldsymbol{\mathcal{Z}}(I_1))$ and then facilitates the readability of the conditional independence hypotheses depicted by the CEG topology.

Second, the order I_1 looks more compelling for domain experts. This is because it makes more sense to describe how a national setting can influence the tendency of their citizens to take cruise trips than the reverse.

Figures 4.3 and 4.4 depict, respectively, the corresponding MAP CEG models for this variable orders I_1 and I_2. Table 4.2 presents the conditional probability table for the MAP CEG $\mathbb{C}(\boldsymbol{\mathcal{Z}}(I_1))$ with a 95% credible interval.

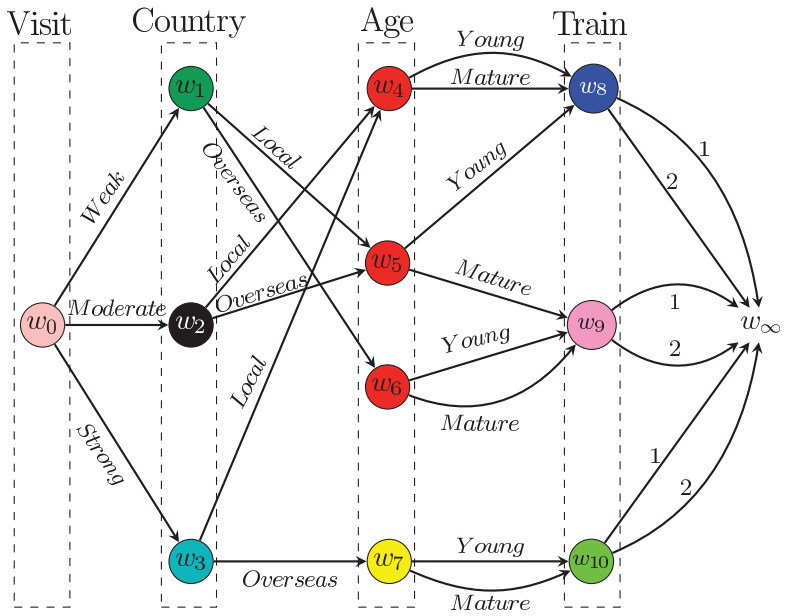


Figure 4.4: The MAP SCEG corresponding to the train booking process when the demographic variables are taken into consideration. Variable order $I_2 = (V, C, A, T)$.

Since positions w_3, w_4 and w_5 in Figure 4.3 have the same colour red (stage u_3) an important context-specific conditional statement stands out. Here the variable Age is independent of variables Country and Visit given that a passenger does not have a strong tendency to travel on cruisers and is not overseas (position w_6). Observe that tourists at the position w_6 have a propensity to be older (78%) than

those passengers at positions w_3, w_4 and w_5 (64%). Recall from Table 3.3 that 67% of tourists in our sample are mature individuals.

Stage	**State Space**	**Mean (95% credible interval)** (%)		
u_0	(Local,Overseas)	57 (52,62)	43 (38,48)	-
u_1	(Weak,Moderate,Strong)	41 (35,47)	38 (32,44)	21 (16,27)
u_2	(Weak,Moderate,Strong)	11 (7,16)	30 (24,37)	59 (52,66)
u_3	(Young,Mature)	36 (31,42)	64 (58,69)	-
u_4	(Young,Mature)	22 (14,30)	78 (70,86)	-
u_5	(1,2)	64 (57,71)	36 (29,43)	-
u_6	(1,2)	44 (35,54)	56 (46,65)	-
u_7	(1,2)	24 (16,32)	76 (68,84)	-

Table 4.2: Posterior mean and 95% credite intervals for the stages corresponding to the MAP CEG $\mathbb{C}(\boldsymbol{Z}(I_1))$ with demographic variables depicted in Figure 4.3.

The MAP CEG model in Figure 4.3 also shows that variable Train is conditionally independent of variable Age given that the variables Country and Visit are known and a tourist is not at position w_4: he is not a local or overseas tourist with, respectively, a weak or moderate inclination for cruise travels.

According to the train options, the MAP CEG model in Figure 4.3 indicates that tourists can be divided into three categories which correspond to positions w_7, w_8 and w_9. Overseas young passengers with a moderate propensity for cruise trips and local passengers except mature ones with a weak inclination for cruise trips (position w_7) tend to buy local train tickets (64%). On the other hand, most overseas tourists with a strong tendency for taking cruisers (position w_9) have a strong preference (76%) for cruise trains. Other tourists have a more balanced preference between public (44%) and cruise (56%) trains.

Chapter 5

Using Non-Local Priors for CEG Model Selection

This chapter constitutes new work for the thesis which has also already been reported in Collazo and Smith (2016). Here my focus will be the search over the space of CEGs that can also be expressed as context-specific BNs. This enables us to choose priors on hyper-parameters of the different component models so that the higher scoring models tend to be the simpler ones. One such family of priors that has this property is that of the so-called Non-Local Priors (NLPs).

The present chapter begins with a brief review of NLP distributions and all the following sections comprise entirely new developments of NLPs applied to CEGs that I have developed for this thesis. I proceed to discuss some undesirable phenomena that can stack up to occur when Dirichlet prior distributions are used for CEG model selection. To circumvent these issues I will then propose three new families of NLPs for discrete process represented by tree-based graphical models: the full product NLPs (fp-NLPs), the pairwise product NLPs (pp-NLPs) and the pairwise moment NLPs (pm-NLPs). Although these methods are developed for CEG models, I have noted that they can also be directly extended to other applications, for example, to Bayesian cluster analyses.

I discover that the great advantage of a pm-NLP is that it retains the learning

rate associated with more standard priors if the data generating process is the complex model whilst also scaling up the learning rate when the simple model is true. This enforces parsimony over the model selection in a direct and simple way, keeping computational time and memory costs under control. The empirical results presented here also indicate that a CEG model search using pm-NLPs is more robust than one using a standard Dirichlet prior in the sense that model selection is similar for wide intervals of values of nuisance hyper-parameters.

The necessity for heuristic algorithms for CEG model selection has already been stressed in Silander and Leong (2013) and Cowell and Smith (2014). I will also show here that a pm-NLP helps to reduce the incidence of some unwanted properties exhibited by standard Dirichlet local priors or product NLPs (fp-NLPs and pp-NLPs) when these priors are used in conjunction with greedy algorithms. Next I will develop a formal framework that enables us to employ pm-NLPs for CEG model search within my OAHC algorithm described in Section 4.2.1. To show the efficacy of this method, I present extensive computational experiments for CEG model selection associated with health and security applications.

5.1 Introduction to Non-Local Prior Distributions

In the literature, there are some compelling reasons for adopting a Bayesian approach for model selection, which often use BFs based on conjugate priors; see e.g. Berger and Pericchi (2001). These justifications in favour of BFs can be summarized in the following four points:

1. Its comprehension is direct and intuitive, especially for non-specialists, avoiding, for example, common difficulties associated with p-values.
2. It is robust under certain conditions and conceptually consistent. It can also be used to compare non-nested models.
3. It provides us with a posterior probability distribution over the model space. This enables us to perform model averaging instead of basing a decision only on a single model.

4. Its implementation implicitly balances model complexities and data information. This minimizes the problem of overfitting and so advocates the Occam's razor principle. For an extensive analysis of this particular topic, see MacKay (2003).

However, there are also some criticisms of the Bayesian framework, particularly when improper or vague proper priors are used (Rao and Wu, 2001, Berger and Pericchi, 2001, Pericchi, 2005). The main issue associated with an improper prior is that it yields an arbitrary constant that does not cancel out when comparing models in different dimension spaces. In turn, a vague proper prior makes the result depend on the parameter that sets the vagueness of beliefs a priori and so reduces the robustness of the model selection.

To address these drawbacks, a number of solutions have been proposed. These include the Bayesian information criterion of Schwarz (Schwarz, 1978), the intrinsic BF approach (Berger and Pericchi, 1996a,b, 1998, Moreno, 1997, Moreno et al., 1998), the expected-Posterior prior approach (Perez and Berger, 2002) and the fractional BF aprroach (O'Hagan, 1995, 1997, de Santis and Spezzaferri, 1999). For comparisons of these methods, see Berger and Pericchi (2001) and Pericchi (2005). The relation between the frequentist concept of significance level and the Bayesian information criterion is studied in Efron and Gous (2001).

These BF selection techniques tend to use local priors; that is, priors that keep the null model's parameter space nested in the alternative model's parameter space. For example, consider a choice between the null model

$$\mathbb{M}_0 : p(\boldsymbol{x}|\boldsymbol{\theta}, \mathbb{M}_0),$$

such as $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathbb{M}_0)$ and $\boldsymbol{\theta} \in \Theta_0$, and the alternative model

$$\mathbb{M}_1 : p(\boldsymbol{x}|\boldsymbol{\theta}, \mathbb{M}_1),$$

such as $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathbb{M}_1)$ and $\boldsymbol{\theta} \in \Theta_1, \Theta_0 \subset \Theta_1$. An absolutely continuous prior distribution is called a local prior (LP) if $p(\boldsymbol{\theta}|\mathbb{M}_1)$ is strictly positive for all $\boldsymbol{\theta} \in \Theta_0$.

However, some studies (Dawid, 1999, 2011, Johnson and Rossell, 2010) have shown that local priors are prone to cause an imbalance in the training rate since

the evidential support grows exponentially under a true alternative model but only polynomially under a true null model.

To circumvent this phenomenon, BF selection methods based on non-local priors (NLPs) have been successfully developed for linear models (Johnson and Rossell, 2010, 2012) and graphs of Gaussian variables (Consonni and La Rocca, 2011, Consonni et al., 2013, Altomare et al., 2013). Assume again that we want to choose between two models $\mathbb{M}_0$ and $\mathbb{M}_1$ as described above. An absolutely continuous prior is said to be a non-local prior if and only if for all values $\boldsymbol{\theta}_0 \in \Theta_0$ associated with the null model $\mathbb{M}_0$ we have that

$$\lim_{\boldsymbol{\theta} \to \boldsymbol{\theta}_0} p(\boldsymbol{\theta}|\mathbb{M}_1) = 0. \tag{5.1}$$

An NLP implies that a priori the probability measure of the parameter $\boldsymbol{\theta}$ associated with the alternative model $\mathbb{M}_1$ ($\boldsymbol{\theta} \in \Theta_1$) goes to zero as the value of $\boldsymbol{\theta}$ gets closer to the null model's parameter space Θ_0. Now take a space $\Theta_1(d)$ defined by all points of Θ_1 whose distance to Θ_0 is smaller than d. It then follows that under the alternative model the prior probability mass corresponding to $\Theta_1(d)$ is greater than any given ϵ, $0 < \epsilon < 1$, if and only if the distance d is greater than some $\delta, \delta > 0$. This enables an NLP to incorporate a notion of separation between two nested models directly into the prior distribution. In doing this, it does not modify the learning rate under the true alternative model whilst it scales it up under the true null model.

Some previous work used truncated probability measures to accommodate a separation between nested models (Verdinelli and Wasserman, 1996, Klugkist and Hoijtink, 2007, Rousseau, 2007). In a setting with two models, this type of truncated priors guarantees that $p(\boldsymbol{\theta} \in \mathcal{N}(\Theta_0))|\mathbb{M}_1)$ is zero for all $\boldsymbol{\theta} \in \Theta_0$, where $\mathcal{N}(\Theta_0), \mathcal{N}(\Theta_0) \subset \Theta_1$, is some non-zero Lebesgue measure neighbourhood of θ_0 in the parameter space of model $\mathbb{M}_1$.

However enforcing a sharp transition from a region of positive probability distribution to a null one, these truncated priors have two problems that an NLP does not have (Johnson and Rossell, 2010, Rossell and Telesca, 2015). They do not allow

us to set the rate that $p(\boldsymbol{\theta}|\mathbb{M}_1)$ goes to zero as $\theta, \theta \in \Theta_1$, gets closer to Θ_0 in the parameter space of the containing model $\mathbb{M}_1$. In addition, they yield a lack of consistency between the parameter estimation and a hypothesis test. For a useful characterization of NLPs as a mixture of truncated distributions, see Rossell and Telesca (2015).

5.2 Introduction to Non-local Priors for CEGs

For CEG model selection, we need to determine when it is better to hold situations apart or merge these into a single stage. The standard BF score can induce rather strange optimal combinations of stages, when the compared stages have very different visit rate ($\bar{\phi}_i$). Theorem 2 below provides us the asymptotic form of $lpBF$ using Dirichlet local priors and makes explicit why difficulties can arise in this context.

Let $\boldsymbol{\phi_i} = (\phi_{i1}, \ldots, \phi_{iL_i})$ denote a vector whose element ϕ_{ij} corresponds to the probability of an individual arriving at a stage u_i and taking the emanating edge j of u_i. Then clearly $\phi_{ij} = \bar{\phi}_i \times \pi_{ij}$. Note that the visit rate $\bar{\phi}_i$ can be formally defined in terms of the path $\sigma-$algebra (Smith and Anderson, 2008) yielded by a CEG $\mathbb{C}$ as follows:

$$\bar{\phi}_i = p(\Lambda(u_i)) \quad \text{and} \quad \bar{\phi}_{ij} = p(\Lambda_j(u_i)),$$

where $\Lambda(u_i)$ is the set of all paths in $\mathbb{C}$ that pass through at least one position in the stage u_i and where $\Lambda_j(u_i)$ is the subset of all paths in $\Lambda(u_i)$ that pass through an edge j corresponding to the stage u_i.

So each stage u_i can be associated with a random variable $\Phi_i \sim Bernoulli(\bar{\phi}_i)$ that represents whether or not an individual visits that stage. Analogously each emanating edge j of a stage u_i can be linked to the level of a random variable $\Phi_{ij} \sim Bernoulli(\phi_{ij})$ representing whether or not an individual arrives at u_i and takes that edge j. Recall from Section 4.1.3 that in applied studies standard CEG model selection requires us to assume complete random sampling and so these

random variables are valid even for a CEG where a path passes through the same stage two or more times. Of course, this assumption should be assured by careful experimental design.

Theorem 2. *Take two CEGs $\mathbb{C}$ and $\mathbb{C}^+$ such as $\mathbb{C}^+$ is 1-nested in $\mathbb{C}$. Assume that stages $u_1, u_2 \in \mathbb{C}$ are merged into the stage $u_{1\oplus 2} \in \mathbb{C}^+$. Consider also the true positive conditional probabilities $\boldsymbol{\pi}_1^\dagger$ and $\boldsymbol{\pi}_2^\dagger$ as well as the true positive probabilities $\boldsymbol{\phi}_1^\dagger$ and $\boldsymbol{\phi}_2^\dagger$ associated with stages u_1 and u_2, respectively. If both CEGs have the same prior distribution over the model space $\boldsymbol{\mathcal{C}}$ (see Section 4.1.2), then as $n \to \infty$*

$$lpBF[\mathbb{C}, \mathbb{C}^+] \xrightarrow{a.s.} nB(\boldsymbol{\pi}_1^\dagger, \boldsymbol{\pi}_2^\dagger, \boldsymbol{\phi}_1^\dagger, \boldsymbol{\phi}_2^\dagger) - \frac{L-1}{2} log(n) + A(\boldsymbol{\phi}_1^\dagger, \boldsymbol{\phi}_2^\dagger, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2), \quad (5.2)$$

where A and B are constants that depend on their arguments as given above, and n is the sample size.

Proof. Using the fact that $\ln \Gamma(z) = (z-0.5) \times \ln(z) - z + 0.5 \times \ln(2\pi) + O(1)$ as $z \to \infty$ (Abramowitz and Stegun, 1972), we can rewrite equation 4.8 as follows

$$\begin{aligned} lpBF(\mathbb{C}, \mathbb{C}^+) = & \sum_{i=1}^{L_1} \alpha_{1i}^* \ln\left(\frac{\alpha_{1i}^*}{\alpha_{1i}^* + \alpha_{2i}^*} \frac{\bar{\alpha}_1^* + \bar{\alpha}_2^*}{\bar{\alpha}_1^*}\right) \\ & + \sum_{i=1}^{L_1} \alpha_{2i}^* \ln\left(\frac{\alpha_{2i}^*}{\alpha_{1i}^* + \alpha_{2i}^*} \frac{\bar{\alpha}_1^* + \bar{\alpha}_2^*}{\bar{\alpha}_2^*}\right) \\ & + \frac{1}{2} \ln\left(\frac{\bar{\alpha}_1^* \bar{\alpha}_2^*}{\bar{\alpha}_1^* + \bar{\alpha}_2^*}\right) - \frac{1}{2} \sum_{i=1}^{L_1} \ln\left(\frac{\alpha_{1i}^* \alpha_{2i}^*}{\alpha_{1i}^* + \alpha_{2i}^*}\right) \\ & + A(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) + O(1). \end{aligned} \quad (5.3)$$

Using the Strong Law of Large Numbers and the continuous mapping theorem (Billingsley (1999)), we obtain that as $n \to \infty$

$$\begin{aligned} & lpBF(\mathbb{C}, \mathbb{C}^+) \xrightarrow{a.s.} \\ & n\left\{ \sum_{i=1}^{L_1} \phi_{1i}^\dagger \ln\left[\pi_{1i}^\dagger \left(\frac{\bar{\phi}_1^\dagger + \bar{\phi}_2^\dagger}{\pi_{1i}^\dagger \bar{\phi}_1^\dagger + \pi_{2i}^\dagger \bar{\phi}_2^\dagger}\right)\right] + \sum_{i=1}^{L_1} \phi_{2i}^\dagger \ln\left[\pi_{2i}^\dagger \left(\frac{\bar{\phi}_1^\dagger + \bar{\phi}_2^\dagger}{\pi_{1i}^\dagger \bar{\phi}_1^\dagger + \pi_{2i}^\dagger \bar{\phi}_2^\dagger}\right)\right] \right\} \\ & - \frac{L-1}{2} \ln(n) - \frac{1}{2} \ln\left(\frac{1}{\bar{\phi}_1^\dagger} + \frac{1}{\bar{\phi}_2^\dagger}\right) + \frac{1}{2} \sum_{i=1}^{L_1} \ln\left(\frac{1}{\phi_{1i}^\dagger} + \frac{1}{\phi_{2i}^\dagger}\right) + A(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2). \end{aligned} \quad (5.4)$$

■

Note that the evidence in favour of any model depends on the sign of the constant B that is analysed in the next two corollaries. As expected, Corollary 1 tells us that there is an imbalance between the learning rates of simple and complex models since the evidence grows logarithmically if the true model is the simple one and linearly otherwise.

Corollary 1. *Take two CEGs $\mathbb{C}$ and $\mathbb{C}^+$ as defined in Theorem 2. If $\boldsymbol{\pi}_1^\dagger = \boldsymbol{\pi}_2^\dagger$, then $B = 0$.*

Proof. This follows directly from equation 5.4. ■

Corollary 2 tells us that in any agglomerative search those stages that are more likely to be visited tend to attract stages that are only visited rarely. This is regardless of the generating processes that characterises the conditional probability distributions of these stages.

Corollary 2. *Take two CEGs $\mathbb{C}$ and $\mathbb{C}^+$ as defined in Theorem 2. Consider $\boldsymbol{\phi}_2^\dagger = \kappa \boldsymbol{\phi}_1^\dagger$ where κ is a positive real constant and $\boldsymbol{\pi}_1^\dagger \neq \boldsymbol{\pi}_2^\dagger$. Then, for sufficiently small κ, $B < 0$ regardless of the true conditional probabilities $\boldsymbol{\pi}_1^\dagger$ and $\boldsymbol{\pi}_2^\dagger$.*

Proof. Assume $D_{KL}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2,)$ as the Kullback-Leibler divergence between the discrete probability distributions $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Using $\bar{\phi}_2^\dagger = \kappa \bar{\phi}_1^\dagger$ and $\ln(1+z) = z + O(z^2)$ as $z \to 0$, we can rewrite B as follows:

$$\begin{aligned} B &= \bar{\phi}_1^\dagger \left\{ (\kappa+1)\ln(\kappa+1) - \sum_{i=1}^{L_1} \left[\pi_{1i}^\dagger \ln\left(1 + \kappa \frac{\pi_{2i}^\dagger}{\pi_{1i}^\dagger}\right) + \kappa \pi_{2i}^\dagger \ln\left(\kappa + \frac{\pi_{1i}^\dagger}{\pi_{2i}^\dagger}\right) \right] \right\} \\ &= \bar{\phi}_1^\dagger \left[O(\kappa^2) - \kappa \sum_{i=1}^{L_1} \pi_{2i}^\dagger \ln\left(\kappa + \frac{\pi_{1i}^\dagger}{\pi_{2i}^\dagger}\right) \right] \\ &\leq \bar{\phi}_1^\dagger \left[O(\kappa^2) - \kappa \sum_{i=1}^{L_1} \pi_{2i}^\dagger \ln \frac{\pi_{1i}^\dagger}{\pi_{2i}^\dagger} \right] = \bar{\phi}_1^\dagger \left[O(\kappa^2) - \kappa D_{KL}(\boldsymbol{\pi}_2^\dagger, \boldsymbol{\pi}_1^\dagger) \right]. \end{aligned} \tag{5.5}$$

Note that the inequality holds because κ is strictly positive. The result follows since the Kullback-Leibler divergence is always non-negative and is equal to 0 if and only if $\boldsymbol{\pi}_1^\dagger = \boldsymbol{\pi}_2^\dagger$. ■

Define the distance between any two stages as given by the distance between their associated expected floret edge probabilistic vectors. According to Corollary 3 massive (or often visited) stages tend to attract to them very light (or less visited) ones no matter how far away these other light stages are in the probabilistic space. Obviously this is not ideal for highly separated stages to be combined together: they clearly make very different predictions about what will happen to a unit arriving there. Corollary 3 also shows that in contrast, even if other massive stages are very close to each other and so natural to combine, these stages will be less prone to be amalgamated together than in the previous case. Although this is a familiar problem in classical hypothesis testing where statistically different hypotheses might not be significantly different from an interpretative viewpoint, this is nevertheless not a desirable property for Bayesian search algorithms.

Corollary 3. *Take three CEGs $\mathbb{C}$, $\mathbb{C}_1^+$ and $\mathbb{C}_2^+$ where $\mathbb{C}_1^+$ and $\mathbb{C}_2^+$ are 1-nested in $\mathbb{C}$. Assume also that the CEG $\mathbb{C}^\dagger$ is the true model and that this is m-nested in the CEG $\mathbb{C}_1^+$ but is not nested in the CEG $\mathbb{C}_2^+$. If the two stages we combine in CEG $\mathbb{C}$ to form a CEG $\mathbb{C}_2^+$ fulfil the conditions of Corollary 2, then as $n \to \infty$*

$$lpBF[\mathbb{C}, \mathbb{C}_2^+] - lpBF[\mathbb{C}, \mathbb{C}_1^+] \xrightarrow{a.s.} nB_2 + A_2 - A_1 \tag{5.6}$$

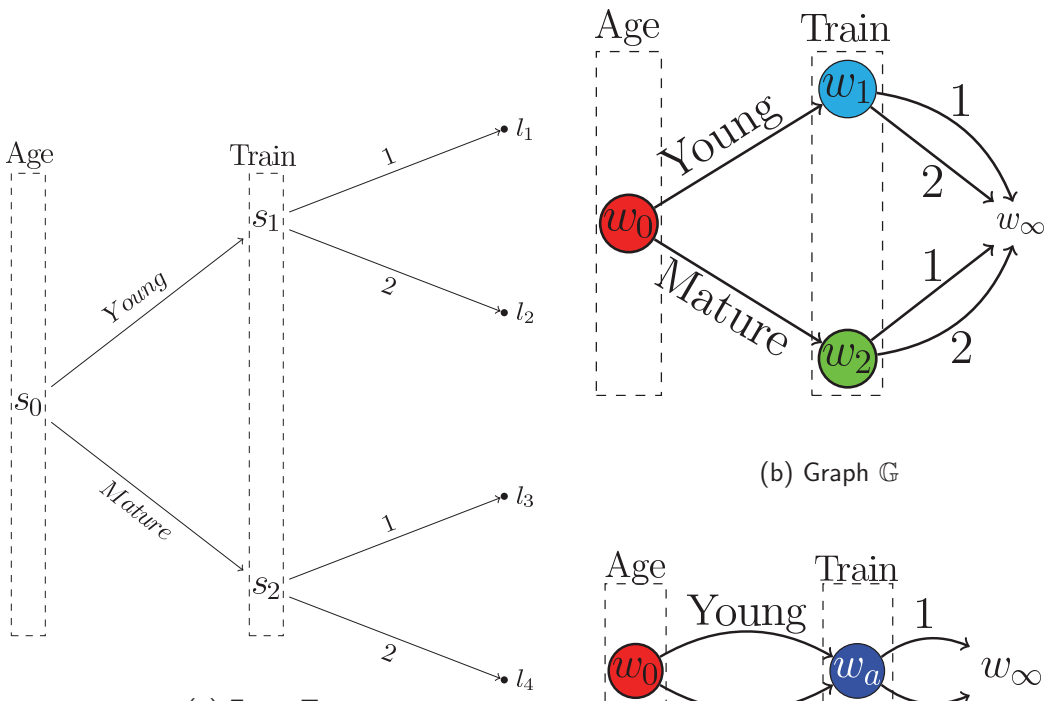
where A_1 is a constant as defined in equation 5.2 for SCEGs $\mathbb{C}$ and $\mathbb{C}_1^+$, A_2 and B_2 are the corresponding constants given in equation 5.2 for CEGs $\mathbb{C}$ and $\mathbb{C}_2^+$ and where $B_2 < 0$.

Proof. The result follows directly from Corollaries 1 and 2. ■

So in this sense the standard BF score can lead to poor model choice when a *pairwise selection* process like the AHC algorithm is used with Dirichelet local priors. The AHC algorithm, which is based on such a sequence of pairwise selection steps, can therefore be sometimes led away from selecting an appropriate model. In fact this phenomenon is actually exacerbated because of the sequential nature of the AHC algorithm. Once a stage with high true visit rate attracts erroneously other less visited stages, it becomes more massive and therefore more prone to gather

incorrectly other smaller stages as the AHC algorithm sequentially agglomerates situations.

The NLP becomes a good option to circumvent this issue. It does this by introducing a formal measure of separation between partitions of the model. This ensures the selection of models not only depends on the probability mass of their partitions but also on the relative distances between their associated probability measures. NLPs therefore provide a promising generic method to more appropriately score CEGs for two main reasons. These priors reduce the imbalance in the learning rate and enforce parsimony in the model selection. They also discourage a greedy model search algorithm from merging two stages spuriously simply because of the probability mass effects discussed above.



(a) Event Tree

(b) Graph $\mathbb{G}$

(c) Graph $\mathbb{G}^+$

Figure 5.1: An Event Tree and two possible CEGs that can be modelled using the Train Booking example (see Section 3.1) with *only two* variables, Age and Train.

To illustrate how we might construct NLPs for CEGs, consider only two variables

of the train booking example (Section 3.1), Age and Train. The corresponding event tree of this process is presented in Figure 5.1a.

Here it is only possible to obtain one of two graphs: $\mathbb{G}$ with two different stages $u_1 = \{w_1\} = \{s_1\}$ and $u_2 = \{w_2\} = \{s_2\}$ as presented in Figure 5.1b; or graph $\mathbb{G}^+$ with only one stage $u_a = \{w_a\} = \{s_1, s_2\}$ as presented in Figure 5.1c, where the stages u_1 and u_2 of $\mathbb{G}$ are merged into a single stage u_a. During the model selection, we need to test whether the stages u_1 and u_2 should be merged or not: $H_0 : \pi_1 = \pi_2$ *vs* $H_1 : \pi_1 \neq \pi_2$.

To do this I construct NLPs that combine the distance between these two stages $d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ and their probability densities yielded by standard Dirichlet local priors $q_{LP}(\boldsymbol{\pi}_1)$ and $q_{LP}(\boldsymbol{\pi}_2)$. For exemplification, I use the Euclidean, Minkowski and Hellinger distances. The Minkowski distance corresponds to a generalisation of the Euclidean distance to the τ-norm space $\mathcal{L}_\tau$ ($\tau = 1, 2, \ldots$). For two points $S^M = (s_1^M, \ldots, s_n^M) \in \mathbb{R}^n$ and $T^M = (t_1^M, \ldots, t_n^M) \in \mathbb{R}^n$, this is given by

$$d(S^M, T^M) = \|S^M - T^M\|_\tau = \left(\sum_{i=1}^{n} |s_i^M - t_i^M|^\tau \right)^{\frac{1}{\tau}}, \tag{5.7}$$

where $\| \cdot \|_\tau$ is the τ-norm. See Kruskal (1964) for more details. Note that we have the Euclidean distance when $\tau = 2$. For two discrete probability distributions $S^H = (s_1^H, \ldots, s_n^H) \in \mathbb{R}^n$ and $T^H = (t_1^H, \ldots, t_n^H) \in \mathbb{R}^n$, the Hellinger distance (Rao (1995)) is defined by

$$d(S^H, T^H) = \|\sqrt{S^H} - \sqrt{T^H}\|_2 = \left(\sum_{i=1}^{n} (\sqrt{s_i^H} - \sqrt{t_i^H})^2 \right)^{\frac{1}{2}}. \tag{5.8}$$

This can be extend to the 2τ-norm space ($\tau = 1, 2, \ldots$) using the formula

$$d(S^H, T^H) = \|\sqrt[2\tau]{S^H} - \sqrt[2\tau]{T^H}\|_{2\tau} = \left(\sum_{i=1}^{n} (\sqrt[2\tau]{s_i^H} - \sqrt[2\tau]{t_i^H})^{2\tau} \right)^{\frac{1}{2\tau}}. \tag{5.9}$$

An NLP for the stage u_a of $\mathbb{G}^+$ is equal to its Dirichlet local prior since this stage can not be combined with any other stage: $q_{NLP}(\boldsymbol{\pi}_a|\mathbb{G}^+) = q_{LP}(\boldsymbol{\pi}_a)$ (Figure 5.2). The NLP density for stages u_1 and u_2 of $\mathbb{G}$ is given by:

$$q_{NLP}(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2|\mathbb{G}) = \frac{1}{K} d(\pi_1, \pi_2)^{2\rho} q_{LP}(\boldsymbol{\pi}_1) q_{LP}(\boldsymbol{\pi}_2), \tag{5.10}$$

where the proportionality constant $K = E_{\boldsymbol{\pi}_1,\boldsymbol{\pi}_2}[d(\pi_1,\pi_2)^{2\rho}]$ can be calculated simply using the Dirichlet local priors π_1 and π_2 (Figure 5.3).

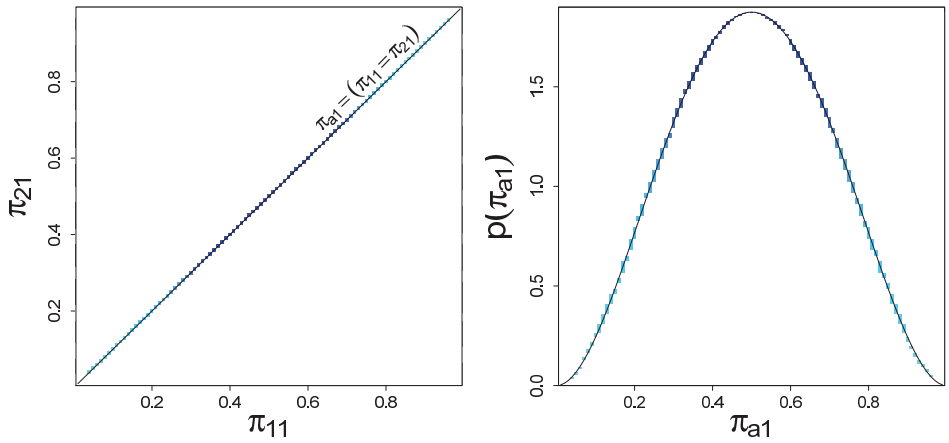
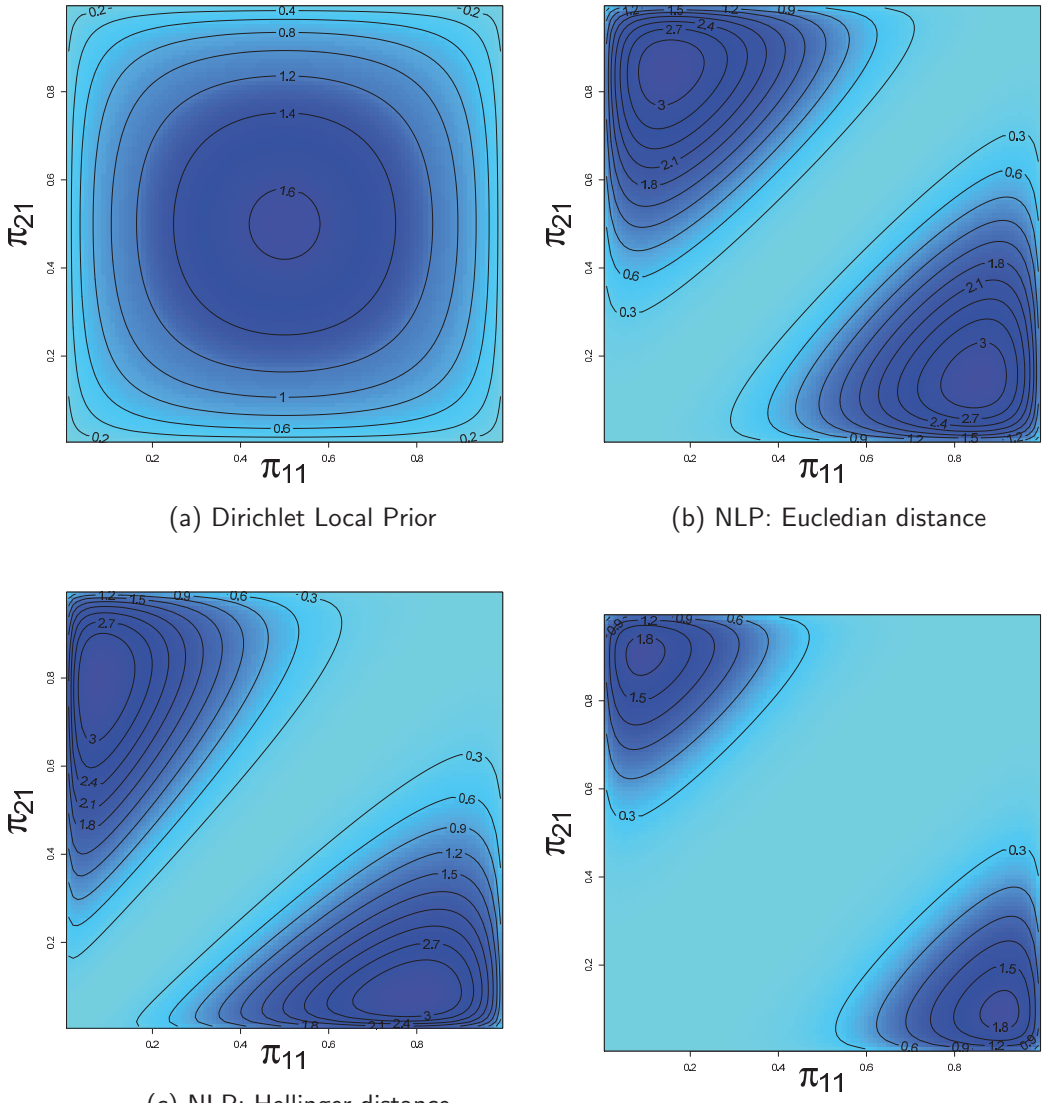


Figure 5.2: NLP coincident with Dirichlet Local Prior for the only stage associated with the variable Train in the graph $\mathbb{G}^+$ depicted in Figure 5.1c where $\boldsymbol{\pi}_a \sim Beta(3,3)$ and $\bar{\alpha} = 6$. Deeper colour represents higher probability densities.

Note that the NLP for graph $\mathbb{G}$ (Equation 5.10) vanishes when the cell probability vectors associated with the stages u_1 and u_2 are close to one another (Figures 5.3b, 5.3c, 5.3d). Here the probability mass is concentrated a priori in the probability space where the conditional probabilities $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ are different. This inhibits the NLP in Equation 5.10 for the complex model $\mathbb{G}$ from representing the same stage structure ($\pi_1 = \pi_2$) which is embedded into the simple model $\mathbb{G}^+$. So, NLPs only allow the parameters corresponding to stages u_1 and u_2 to be identified with each other under the null hypothesis H_0.

This contrasts with standard Dirichlet local priors that concentrate the probability mass associated with stages u_1 and u_2 of $\mathbb{G}$ around the probability space where these parameters are equal (Figure 5.3a). In this sense, local priors do not establish a full partition of the parameter space: the null hypothesis H_0 is nested into the graph $\mathbb{G}$ that should represent only the hypothesis H_1. When using NLPs, these two stages will remain separated or not, based not only on their consistency with the data but also on how far apart these models are, as measured by the distances defined above. Thus as the basis of moderate amount of data, situations tend



(a) Dirichlet Local Prior

(b) NLP: Eucledian distance

(c) NLP: Hellinger distance

(d) NLP: Minkowski distance ($\mathcal{L}_4$)

Figure 5.3: Dirichlet Local Prior and NLPs using different distances for stages associated with the variable Train in the graph $\mathbb{G}$ depicted in Figure 5.1b where $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2 \sim Beta(1.5, 1.5)$ and $\bar{\alpha} = 6$. Deeper colour represents higher contours.

to be placed in the same stage (graph $\mathbb{G}^+$) unless their edge probabilities are sufficiently different (graph $\mathbb{G}$). We will see at the end of this Chapter that this enables us to discover models admitting parsimonious explanations as well as good fits to the data.

Remember that we need to elicit a prior joint distribution $p(\boldsymbol{\pi}, \mathbb{G})$ to embed a probabilist map into CEG models. Using Dirichlet local priors and the usual conventions (see e.g. Heckerman (1999)), the parameter $\boldsymbol{\pi}$ and the graph $\mathbb{G}$ are

mutually independent a priori: $p(\boldsymbol{\pi}, \mathbb{G}) = p(\boldsymbol{\pi})p(\mathbb{G})$. This does not happen with NLPs since the prior distribution over the parameter space is conditional on the graph $\mathbb{G}$: $p(\boldsymbol{\pi}, \mathbb{G}) = p(\boldsymbol{\pi}|\mathbb{G})p(\mathbb{G})$. Observe in Figure 5.3 that given a prior distribution $p(\mathbb{G})$ NLPs reduce the density $p(\boldsymbol{\pi}, \mathbb{G})$ in comparison to local priors only when the distances between the parameters in the corresponding CEGs are close. In contrast, when these distances are substantially different from zero the density indeed increases. In this way, NLPs bias the CEG model selection towards simpler models but only when the data supports them.

Of course, although for simplicity I do not consider this possibility here, I could choose to impose a prior over the model space that further favoured parsimonious models. Note however that although non-uniform priors over the CEG model space reduce the density $p(\boldsymbol{\pi}, \mathbb{G})$ of complex models they do this regardless of the data generation processes. In these cases, the biases in favour of simpler models need to be based on some prior "objective" hypotheses or important prior subjective beliefs over the model space. Despite often being very important in applied studies, these prior distributions are also usually very domain specific. So they are not the focus of this thesis.

5.3 Three new families of NLPs for tree-based models

To extend the previous method of construction of an NLP to the case when there are more than 2 stages (for example, the third level of the CEG in Figure 5.4), a natural option is to take the product distance between the conditional probability distributions for every pair of stages that can be merged. This family of NLPs is consistent in the sense that their constructions only depend on the characteristics of the particular model associated with that prior. Johnson and Rossell (2012) successfully adopted such a product moment NLP (pMOM-NLP) for Bayesian selection in the context of linear regression. We formally define the fp-NLPs for CEGs below.

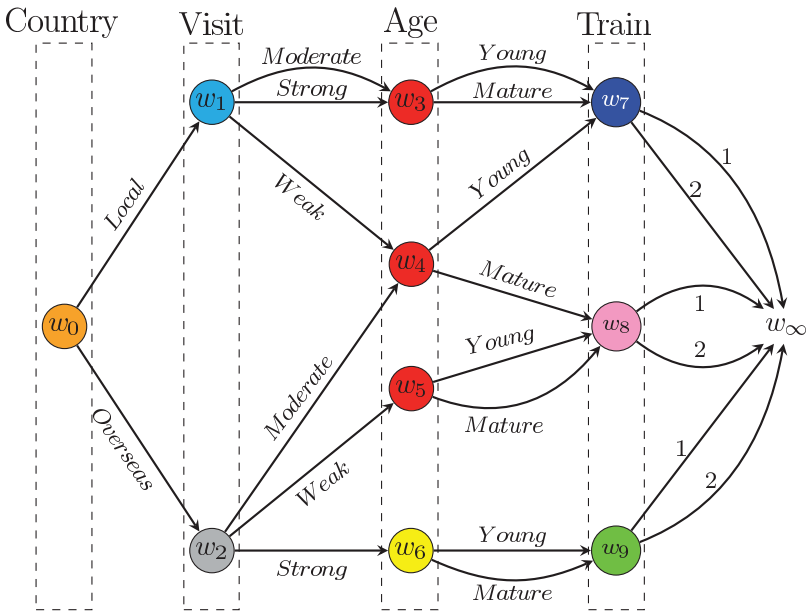


Figure 5.4: The MAP SCEG corresponding to the train booking process (Sections 3.1, 3.2 and 4.5) when the demographic variables are taken into consideration assuming the variable order $C \succ V \succ A \succ T$. The stage structure is given by: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3, w_4, w_5\}$, $u_4 = \{w_6\}$, $u_5 = \{w_7\}$, $u_6 = \{w_8\}$, $u_7 = \{w_9\}$. This CEG is identical to one depicted in Figures 3.3 and 4.3.

In this section, I let $\mathscr{P}_{DLP}$ and $\mathscr{P}_{NLP}$ denote probability measures yielded, respectively, by Dirichlet local priors and NLPs. We also assume that the expectations $E_{\boldsymbol{\pi}}[f(\boldsymbol{\pi})]$ and $E_{\boldsymbol{\pi}*}[f(\boldsymbol{\pi})]$ are calculated, respectively, using the Dirichlet local prior and its corresponding posterior (see Section 3.3) on $\boldsymbol{\pi}$. Finally, in a CEG whose graphical structure associated with a hyper-stage $\mathcal{H}$ is given by $\mathbb{G} = (\mathcal{T}, U)$ I let $\Psi(U)$ denote the collection of pairs of stages (u_i, u_j) in U that can be merged to derive nested CEGs.

To better understand $\Psi(U)$, it is useful to rewrite this as a collection of sets $\{\Psi_k(U)\}_k$, where $\Psi_k(U) = \{u_{r_i}\}_i$ denotes the largest set of stages in U yielded by $\mathcal{H}$ such that the following property holds: for any pair of stages u_{r_1} and u_{r_2} in $\Psi_k(U)$, $(u_{r_1}, u_{r_2}) \in \Psi(U)$. Recall from Section 4.2.1 that a hyper-stage $\mathcal{H}$ does not need to be a partition of the set of situations of $\mathcal{T}$. So, $\{\Psi_k(U)\}_k$ does not also need to be a partition of U, although this property is usually desirable in real-world applications because it simplifies the implementation of model search

algorithms. Now if this property holds, we can write

$$\prod_{\{(u_i,u_j)\}\in\Psi(U)} d(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)^{2\rho} = \prod_{k=1}^{J}\prod_{i=1}^{J_k-1}\prod_{j=2}^{J_k} d(\boldsymbol{\pi}_{r_i},\boldsymbol{\pi}_{r_j})^{2\rho}, \tag{5.11}$$

where $J = |\Psi(U)|$ and $J_k = |\Psi_k(U)|$.

To illustrate this construction take the CEG depicted in Figure 5.4. In this case, the hyper-stage $\mathcal{H}$ is defined according the collection of variables that characterise the process and so is $\Psi(U)$. Thus, in the notation above, we then have that $\Psi(u) \equiv \{\Psi_0 = \{u_0\}, \Psi_1 = \{u_1, u_2\}, \Psi_2 = \{u_3, u_4\}, \Psi_3 = \{u_5, u_6, u_7\}\}$. Here all stages that are associated with the same variable are gathered into the same set $\Psi_k(U)$. For instance, the set Ψ_2 is made up of those stages associated with the variable Age. Note that the positions w_3, w_4 and w_5 are in the same stage u_3.

Definition 26 (Full Product Non-local Priors for CEGs)**.** The fp-NLPs for a CEG $\mathbb{D} = (\mathcal{T}, U, \mathscr{P}_{NLP})$ where $\mathbb{G} = (\mathcal{T}, U)$ and $\Psi(U) \neq \emptyset$ are given by

$$q_{NLP}(\boldsymbol{\pi}|\mathbb{G}) = \frac{1}{K}\left[\prod_{(u_i,u_j)\in\Psi(U)} d(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)^{2\rho}\right] q_{DLP}(\boldsymbol{\pi}|\mathbb{G}), \tag{5.12}$$

where $\rho \in \mathbb{N}^+$ and $K = E_{\boldsymbol{\pi}}\left[\prod_{(u_i,u_j)\in\Psi(U)} d(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)^{2\rho}\right]$ is the normalisation constant. If $\Psi(U)$ is empty then $q_{NLP}(\boldsymbol{\pi}|\mathbb{G}) = q_{DLP}(\boldsymbol{\pi}|\mathbb{G})$.

Assuming random sampling and a non-empty $\Psi(U)$, we can now write the joint distribution of the CEG $\mathbb{D} = (\mathcal{T}, U, \mathscr{P}_{NLP})$ using fp-NLPs as function of the CEG $\mathbb{C} = (\mathcal{T}, U, \mathscr{P}_{DLP})$. Thus:

$$\begin{aligned} p_{NLP}(\boldsymbol{x},\boldsymbol{\pi}|\mathbb{G}) &= p(\boldsymbol{x}|\boldsymbol{\pi},\mathbb{G}) q_{NLP}(\boldsymbol{\pi}|\mathbb{G}) \\ &= p(\boldsymbol{x}|\boldsymbol{\pi},\mathbb{G})\left[\frac{1}{K}\prod_{(u_i,u_j)\in\Psi(U)} d(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)^{2\rho}\right] q_{DLP}(\boldsymbol{\pi}|\mathbb{G}) \\ &= \left[\frac{1}{K}\prod_{(u_i,u_j)\in\Psi(U)} d(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)^{2\rho}\right] p_{DLP}(\boldsymbol{x},\boldsymbol{\pi}|\mathbb{G}), \end{aligned} \tag{5.13}$$

So, we have that

$$p_{NLP}(\boldsymbol{\pi}|\boldsymbol{x},\mathbb{G}) = \left[\frac{1}{K^*}\prod_{(u_i,u_j)\in\Psi(U)} d(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)^{2\rho}\right] p_{DLP}(\boldsymbol{\pi}|\boldsymbol{x},\mathbb{G}), \tag{5.14}$$

where $K^* = E_{\boldsymbol{\pi}^*}\left[\prod_{(u_i,u_j)\in\Psi(U)} d(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)^{2\rho}\right]$ is the normalisation constant. After a little algebra this can be rearranged as

$$p_{NLP}(\boldsymbol{x}|\mathbb{G}) = \frac{K^*}{K} p_{DLP}(\boldsymbol{x}|\mathbb{G}). \tag{5.15}$$

In this case, the lpBF between two CEGs $\mathbb{D}_1$ and $\mathbb{D}_2$ that have the same prior probability over the model space is given by

$$lpBF(\mathbb{D}_1, \mathbb{D}_2) = lpBF(\mathbb{C}_1, \mathbb{C}_2) + \ln K_1^* - \ln K_2^* - \ln K_1 + \ln K_2, \tag{5.16}$$

where $\mathbb{C}_1$ and $\mathbb{C}_2$ are the CEGs using Dirichlet local priors that correspond to CEGs $\mathbb{D}_1$ and $\mathbb{D}_2$ using fp-NLPs, respectively. Note that $K = K^* = 1$ if $\Psi(U)$ is empty.

In view of the large size of the CEG space that grows in terms of the Bell number, to develop efficient search algorithms it is important to keep calculations as simple as possible, and preferably in closed form. One of the easiest way to do this is to use the Euclidean distance in the formulae above and to set $\rho = 1$. We can also impose a further simplifying condition that $\Psi(U)$ is a partition of the stage set U. But even then in this simple case, for each set $\Psi_k(U) \in \Psi(U)$ of a candidate CEG model we need to calculate a mean of the homogeneous symmetric polynomial $\prod_{i=1}^{J_k-1}\prod_{j=i+1}^{J_k} d(\boldsymbol{\pi}_{r_i}, \boldsymbol{\pi}_{r_j})^2$ using the prior and the posterior distributions of the parameters $\boldsymbol{\pi}_i's$. Recall from Section 4.4 that J_k are often very large. The computations of fp-NLPs can therefore quickly become unmanageable as we scale up the number of variables incorporated into an CEG.

There are also other pitfalls when the fp-NLP is used in conjunction with a greedy search algorithm like the AHC. Using a fp-NLP, Theorem 3 below shows us that the normalisation constant of the posterior distribution of $\boldsymbol{\pi}$ converges to zero with probability 1 if there are at least two stages with the same generating processes. This will happen regardless of whether these stages are under assessment by the model search algorithm. In these cases, Theorem 4 tells us that the marginal posterior probability of such CEG also tends to zero with probability 1. Because of this phenomenon, the fp-NLP is often not a good choice when used in conjunction with a sequential greedy model search even though the method encourages a choice of model with a parsimonious graph.

Let $\boldsymbol{Z}^{(n)} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n)$, where the random variable $\boldsymbol{Z}_s$ registers the events that happen to the s^{th} unit in a process supported by an event tree $\mathcal{T}$. Observe that the event tree $\mathcal{T}$ maps $\boldsymbol{z}^{(n)} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$ into a sample $\boldsymbol{x}^{(n)}$ of size n. So, as n increases $\boldsymbol{z}^{(n)}$ yields a sequence of posterior distributions $p(\boldsymbol{\pi}|\boldsymbol{x}^{(n)}, \mathbb{G})$ for the parameter $\boldsymbol{\pi}$. For notational convenience, define a random variable

$$\boldsymbol{\pi}^*(\boldsymbol{Z}^{(n)}) \sim p(\boldsymbol{\pi}|\boldsymbol{X}^{(n)}, \mathbb{G})$$

and let $\boldsymbol{\pi}_i^\dagger = (\pi_{i1}^\dagger, \ldots, \pi_{iL_i}^\dagger)$ be the true conditional probability associated with the stage u_i. For clarity, I sometimes write $K^*(\boldsymbol{Z}^{(n)})$ to emphasize that the normalisation constant of a posterior distribution is determined by a sequence $\{\boldsymbol{Z}^{(n)}, n \geq 1\}$. In this Chapter, the notation $\boldsymbol{z}^{(n)}$ differs from the notation $\boldsymbol{z}^{(k)}$ introduced in Section 4.1.1 to represent an element of the product space $\mathbb{Z}^{(k)}(I)$.

Lemma 1. *Take the probabilistic parameter π_{ij} associated with the emanating edge j of stage u_i with a positive visiting probability in a CEG $\mathbb{C} = (\mathbb{G}, \mathscr{P}_{DLP})$ and consider $\pi_{ij}^\dagger$ its corresponding true parameter. Then, for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we have that for all $\epsilon > 0$*

$$\lim_{n\to\infty} P(|\pi_{ij}^*(\boldsymbol{Z}^{(n)}) - \pi_{ij}^\dagger| > \epsilon) = 0. \tag{5.17}$$

Proof. Using the Strong Law of Large Numbers, it is easy to see that as $n \to \infty$

$$E[\pi_{ij}^*(\boldsymbol{Z}^{(n)})] = \frac{\alpha_{ij}^*}{\bar{\alpha}_i^*} \xrightarrow{a.s.} \pi_{ij}^\dagger, \tag{5.18}$$

and

$$Var[\pi_{ij}^*(\boldsymbol{Z}^{(n)})] = \frac{\alpha_{ij}^*(1 - \frac{\alpha_{ij}^*}{\bar{\alpha}_i^*})}{\bar{\alpha}_i^*(1 + \bar{\alpha}_i^*)} \xrightarrow{a.s.} 0. \tag{5.19}$$

It follows that

$$E_{\pi_{ij}^*(\boldsymbol{Z}^{(n)})}[(\pi_{ij} - \pi_{ij}^\dagger)^2] = Var[\pi_{ij}^*(\boldsymbol{Z}^{(n)})] + (E[\pi_{ij}^*(\boldsymbol{Z}^{(n)})] - \pi_{ij}^\dagger)^2 \xrightarrow{a.s.} 0. \tag{5.20}$$

Since $\pi_{ij}^*(\boldsymbol{Z}^{(n)})$ converges in quadratic means to the true value of the parameter π_{ij} for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$, it also converges in probability to the

true value of the parameter π_{ij} for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$. Note that this result also follows directly from Doob's Theorem (see e.g. Schervish (1996), Section 7.4.1, or DasGupta (2008), Section 20.7). ■

Theorem 3. *Take a continuous and bounded metric d. In a CEG $\mathbb{C} = (\mathbb{G}, \mathscr{P}_{DLP})$ whose conditional probabilities associated with each edge are strictly positive, for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we then have that as $n \to \infty$*

$$E_{\pi^*_{ij}(\boldsymbol{Z}^{(n)})}\left[\prod_{\substack{(u_i,u_j)\\ \in\Psi(U)}} d(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)^{2\rho}\right] \to \prod_{\substack{(u_i,u_j)\\ \in\Psi(U)}} d(\boldsymbol{\pi}^\dagger_i, \boldsymbol{\pi}^\dagger_j)^{2\rho}. \tag{5.21}$$

Proof. This follows directly from Lemma 1 and from the continuous mapping theorem (Billingsley (1999)). ■

Theorem 4. *Let a CEG $\mathbb{C} = (\mathbb{G}, \mathscr{P}_{DLP})$ have conditional probabilities associated with each edge which are strictly positive. Consider the case when at least two stages in $\mathbb{C}$ that can be merged have the same true conditional probability according to a continuous and bounded metric d. For a CEG $\mathbb{D} = (\mathbb{G}, \mathscr{P}_{NLP})$ whose probability measure $\mathscr{P}_{DLP}$ is generated by a fp-NLP, for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we then have that as $n \to \infty$*

$$P(\mathbb{D}|\boldsymbol{X}^{(n)}, \mathbb{G}) \to 0. \tag{5.22}$$

Proof. From Equation 5.15, we have that

$$p(\mathbb{D}|\boldsymbol{x}^{(n)}, \mathbb{G}) = \frac{K^*}{K} p(\mathbb{C}|\boldsymbol{x}^{(n)}, \mathbb{G}). \tag{5.23}$$

As $0 \leq p(\mathbb{C}|\boldsymbol{x}^{(n)}, \mathbb{G}) \leq 1$ and K is a constant that depends on the hyper-parameter $\bar{\alpha}$, we can conclude that

$$\lim_{n\to\infty} K^*(\boldsymbol{z}^{(n)}) = 0 \Rightarrow \lim_{n\to\infty} p(\mathbb{D}|\boldsymbol{x}^{(n)}, \mathbb{G}) = 0. \tag{5.24}$$

Note now that there are at least two stages u_a and u_b in $\mathbb{C}$ that have the same true conditional probability. So, $d(\pi^\dagger_a, \pi^\dagger_b) = 0$ for some pair of stages u_a and u_b

in $\mathbb{C}$. Recall that

$$K^*(\boldsymbol{z}^{(n)}) = E_{\pi^*_{ij}(\boldsymbol{z}^{(n)})}\left[\prod_{\substack{(u_i,u_j)\\ \in\Psi(U)}} d(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)^{2\rho}\right] \tag{5.25}$$

Theorem 3 then implies that Equation 5.24 is always satisfied for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$. ■

Corollary 4 tells us that when the fp-NLP is used the AHC algorithm can misdirect the search since the normalisation constant of the posterior distribution of $\boldsymbol{\pi}$ may vanish even if the separation between stages does not go to zero in the search neighbourhood. This happens because of the interaction between the definition of fp-NLPs and the data generating process: fp-NLPs are constructed using the product distance between every pair of parameters associated with stages that can be merged $(\Psi(U))$. In contrast, the search neighbourhood defined for the AHC algorithm is only a single pair of stages in $\Psi(U)$. Note that the normalisation constant of the prior distribution of $\boldsymbol{\pi}$ remains unaffected in this case since it is only determined by the phantom sample.

Due to its sequential local strategy, the AHC algorithm can then merge stages that yield the best local score even when this merging is not supported by the data generation process. This situation is further exacerbated because of the combinatorial possibilities that can give rise to circumstances similar to those of Corollary 4. We emphasise that this problem occurs because an fp-NLP is used *in conjunction with* a typical local search algorithm that for practical reasons we may be forced to adopt: see the comments above. So this is not an issue intrinsically associated with the *form* of an fp-NLP.

Corollary 4. *Take three CEGs $\mathbb{D}$, $\mathbb{D}_1^+$ and $\mathbb{D}_2^+$ whose probability measures are generated by fp-NLPs using a continuous and bounded metric. Consider that $\mathbb{D}_1^+$ merges the stages u_1 and u_2 of $\mathbb{D}$, $\mathbb{D}_2^+$ merges the stages u_1 and u_3 of $\mathbb{D}$ into a new stage u_a whose distance to any stage of $\mathbb{D}_2^+$ is non-null, and the stages u_3 and u_4 of $\mathbb{D}$ have the same generation process. Assume also that the CEG $\mathbb{D}^\dagger$ is the true model that is 1-nested in CEG $\mathbb{D}_1^+$ but is not nested in CEG $\mathbb{D}_2^+$. Then,*

for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ *we have that as* $n \to \infty$

$$\frac{K_1^*(\boldsymbol{Z}^{(n)})}{K_2^*(\boldsymbol{Z}^{(n)})} \to 0, \tag{5.26}$$

where K_1^* *and* K_2^* *are the normalisation constants with regard to CEGs* $\mathbb{D}_1^+$ *and* $\mathbb{D}_2^+$*, respectively.*

Proof. Since $\mathbb{D}^\dagger$ is 1-nested into $\mathbb{D}_1^+$, Theorem 3 implies that for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we have that

$$\lim_{n\to\infty} K_1^*(\boldsymbol{Z}^{(n)}) = 0. \tag{5.27}$$

On the other hand, $\mathbb{D}_2^+$ does not have stages with equal true conditional probability distribution by construction. Therefore, Theorem 3 also implies that for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we have that

$$\lim_{n\to\infty} K_2^*(\boldsymbol{Z}^{(n)}) = c \neq 0. \tag{5.28}$$

The result then follows directly from Equations 5.27 and 5.28. ■

To sidestep this difficulty, I propose defining NLPs based on pairwise model selection. We note that Consonni and La Rocca (2011) and Altomare et al. (2013) have both used this approach for BN model search. In this framework, the parameters in the contained model have local prior distributions whilst the parameters in the containing model have product NLP distributions. So the choice of prior used in the containing model depends on the contained model. This inconsistency therefore requires a prior specification on the variable order, although in the examples given in this thesis this order does not appear to have a significant impact on later inference. The associated ambiguities are extremely small and in practice the method still seems to work well outside this context. Other than this technical nicety, a search method based on these product NLPs enforces parsimony over our model selection whilst allowing us to explore the local properties of our model space. For the CEG family, I call this NLP the pairwise product NLP (pp-NLP).

Given two CEGs whose stage structures U and U^+ are nested ($U^+ \subset U$), recall that the symbol $\Delta(U, U^+)$ represents the set of stages of U that are merged to

obtain U^+ (Definition 23). Here $\Psi(\Delta(U, U^+))$ denotes the collection of pair of stages (u_i, u_j) in $\Delta(U, U^+)$ that are gathered in U^+. Analogous to $\Psi(U)$, we can rewrite $\Psi(\Delta(U, U^+))$ as a collection of sets $\{\Psi_k(\Delta(U, U^+))\}_k$. Observe that Equation 5.29 depends on which pair of CEGs are under analysis whilst Equation 5.12 is defined in terms of a particular CEG.

To illustrate the nature of $\Psi(\Delta(U, U^+))$, take again the stage structure U of the CEG in Figure 5.4. Consider another CEG whose stage structure U^+ is 3-nested in U in such way that the stages u_1 and u_2 are merged into a stage u_a, and the stages u_5, u_6 and u_7 are combined into a single stage u_b. Then we have that $\Psi(\Delta(U, U^+)) \equiv \{\Psi_1 = \{u_1, u_2\}, \Psi_2 = \{u_5, u_6, u_7\}\}$ for the pair of stage structures U and U^+.

Definition 27 (Pairwise Product Non-local Priors for CEGs)**.** To compare the graphical structure $\mathbb{G}=(\mathcal{T}, U)$ with its m-nested graphical structure $\mathbb{G}^+=(\mathcal{T}, U^+)$, the pp-NLPs for the CEG $\mathbb{D} = (\mathcal{T}, U, \mathscr{P}_{NLP})$ are given by

$$q_{NLP}(\boldsymbol{\pi}|\mathbb{G}) = \frac{1}{K}\left[\prod_{(u_i,u_j)\in\Psi(\Delta(U,U^+))} d(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)^{2\rho}\right] q_{DLP}(\boldsymbol{\pi}|\mathbb{G}), \tag{5.29}$$

where $K = E_{\boldsymbol{\pi}}\left[\prod_{(u_i,u_j)\in\Psi(\Delta(U,U^+))} d(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)^{2\rho}\right]$, $\rho = 1, 2, \ldots$, is the normalisation constant.

It is easy to see that the complexity of pp-NLPs increases with the number m of nested stages. It can also suffer the same problems as fp-NLPs if the heuristic strategy explores model space neighbourhoods that are smaller than m stages. However since our goal is only to develop search methodologies when a NLP is used in conjunction with the AHC algorithm, we need to consider only 1-nested CEGs. In this context the pairwise moment NLP (pm-NLP) works well for CEG model search. Comparing Equations 5.29 and 5.30, we can see that a $|\Delta(U, U^+)| = 1$.

Definition 28 (Pairwise Moment Non-local Priors for CEGs)**.** To compare the graphical structure $\mathbb{G}=(\mathcal{T}, U)$ and its 1-nested graphical structure $\mathbb{G}^+=(\mathcal{T}, U^+)$ such as $\Delta(U, U^+)=\{\{u_1, u_2\}\}$, the pm-NLPs for the CEG $\mathbb{D}=(\mathcal{T}, U, \mathscr{P}_{NLP})$ are

given by

$$q_{NLP}(\boldsymbol{\pi}|\mathbb{G}) = \frac{1}{K} d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)^{2\rho} q_{DLP}(\boldsymbol{\pi}|\mathbb{G}), \tag{5.30}$$

where $K = E_{\boldsymbol{\pi}_1,\boldsymbol{\pi}_2}[d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)^{2\rho}]$ is the normalisation constant and $\rho = 1, 2, \ldots$

The next corollary shows that a pm-NLP will not exhibit the potential misleading behaviour of the AHC algorithm suffered by product NLPs. The problem is avoided because its normalization constant only goes to zero with probability 1 if and only if both merged stages in the contained model have the same generating process. This is because the normalisation constant is defined using exactly the same search neighbourhood as the AHC algorithm - that is, it is a function of densities associated with a single pair of stages. In Corollary 5, $K^* = E_{\boldsymbol{\pi}_1^*,\boldsymbol{\pi}_2^*}[d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)^{2\rho}]$ is the normalisation constant of the joint posterior distribution of stages u_1 and u_2 when a pm-NLP (Definition 28) is used.

Corollary 5. *Take the CEG D presented in Definition 28 where the metric d is continuous and bounded. Then, for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we have that*

$$\lim_{n\to\infty} K^*(\boldsymbol{Z}^{(n)}) = 0 \Leftrightarrow d(\boldsymbol{\pi}_1^\dagger, \boldsymbol{\pi}_2^\dagger) = 0. \tag{5.31}$$

Proof. From Lemma 1 and from the continuous mapping theorem (Billingsley, 1999), for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we have that as $n \to \infty$

$$E_{\boldsymbol{\pi}_1^*(\boldsymbol{Z}^{(n)}),\boldsymbol{\pi}_2^*(\boldsymbol{Z}^{(n)})}\left[d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)^{2\rho}\right] \to d(\boldsymbol{\pi}_1^\dagger, \boldsymbol{\pi}_2^\dagger)^{2\rho} \tag{5.32}$$

Recall that

$$K^*(\boldsymbol{Z}^{(n)}) = E_{\boldsymbol{\pi}_1^*(\boldsymbol{Z}^{(n)}),\boldsymbol{\pi}_2^*(\boldsymbol{Z}^{(n)})}\left[d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)^{2\rho}\right]. \tag{5.33}$$

If the necessary condition (Expression 5.31) is true, for almost all sequences $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots)$ we then have that as $n \to \infty$

$$K^*(\boldsymbol{Z}^{(n)}) \to 0. \tag{5.34}$$

Equations 5.32 and 5.33 then imply that $d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)^{2\rho} = 0$.

Assuming $d(\boldsymbol{\pi}_1^\dagger, \boldsymbol{\pi}_2^\dagger) = 0$, the sufficiency follows again directly from Equations 5.32 and 5.33. ■

Now take a CEG $\mathbb{C} = (\mathcal{T}, U, \mathscr{P}_{DLP})$ and its 1-nested CEG $\mathbb{C}^+ = (\mathcal{T}, U^+, \mathscr{P}_{DLP})$ which aggregates any two stages u_1 and u_2. Consider the CEG $\mathbb{D} = (\mathcal{T}, U, \mathscr{P}_{NLP})$ whose probability measure is yielded by pm-NLPs. Assuming a uniform prior over the staged structure space, it is straightforward to show that

$$lpBF(\mathbb{D}, \mathbb{C}^+) = \ln \frac{K^*}{K} \frac{p_{DLP}(\boldsymbol{x}|\mathbb{G})}{p_{DLP}(\boldsymbol{x}|\mathbb{G}^+)} \frac{q(\mathbb{G})}{q(\mathbb{G}^+)} = \ln K^* - \ln K + lpBF(\mathbb{C}, \mathbb{C}^+). \tag{5.35}$$

Pairwise moment NLPs for CEGs can therefore be interpreted as a penalisation over the alternative staged structure U with respect to the distance between the conditional probability distributions of both stages u_1 and u_2. The AHC algorithm can easily be adjusted to incorporate pm-NLPs since we only need to add a term $(\ln K^* - \ln K)$ to the regular $lpBF$ score. So regardless of their minor global inconsistency, the use of a pm-NLP in conjunction with the AHC algorithm is highly computational efficient and also has good local properties.

Define the map G such as $G_y(x) = 1$, if $y = 0$, and $G_y(x) = x$, if $y > 0$, and then the function

$$f(x, y) = \frac{\Gamma(x + y)}{\Gamma(x)} = G_y((x + y - 1) \cdot (x + y - 2) \cdot \ldots \cdot x) \tag{5.36}$$

where x and y are real and natural numbers, respectively. Also let

$$B(\boldsymbol{\alpha}) = \frac{\prod_{j=1}^n \Gamma(\alpha_j)}{\Gamma(\bar{\alpha})} \tag{5.37}$$

denote the normalization constant for the Dirichlet distribution parametrised by the vector $\boldsymbol{\alpha}$. The following two theorems give K and K^* of equation 5.35 in closed form with regard to the Minkowski distance (Equation 5.7) and with respect to the extension of Hellinger distance to 2ρ-norm spaces (Equation 5.9).

Lemma 2. *Take two random variables Π_1 and Π_2 which have Dirichlet distributions with parameters $\boldsymbol{\alpha}_1 \in \mathbb{R}_+^L$ and $\boldsymbol{\alpha}_2 \in \mathbb{R}_+^L$, respectively. Define a function*

$c(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = \sum_{j=1}^{L} (\pi_{1j}^{1/a} - \pi_{2j}^{1/a})^{2\rho}$ *where* $a > 0$ *and* $\rho = 1, 2, \ldots$ *Then*

$$E[c(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)] = \frac{1}{B(\boldsymbol{\alpha}_1)B(\boldsymbol{\alpha}_2)} \sum_{j=1}^{L} \sum_{h=0}^{2\rho} \left[\binom{2\rho}{h} (-1)^h B(\hat{\boldsymbol{\alpha}}_1^{j,h}) B(\hat{\boldsymbol{\alpha}}_2^{j,h}) \right], \quad (5.38)$$

where

$$\hat{\alpha}_{1k}^{j,h} = \begin{cases} \alpha_{1k} + \frac{2\rho - h}{a} & \text{if } k = j, \\ \alpha_{1k} & \text{if } k \neq j. \end{cases} \qquad \hat{\alpha}_{2k}^{j,h} = \begin{cases} \alpha_{2k} + \frac{h}{a} & \text{if } k = j, \\ \alpha_{2k} & \text{if } k \neq j. \end{cases}$$

Proof. Expanding the function $f(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ by means of the binomial theorem, we then have that:

$$\begin{aligned} &E[f(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)] \\ &\quad = \int_0^1 \sum_{j=1}^{L} (\pi_{1j}^{1/a} - \pi_{2j}^{1/a})^{2\rho} \frac{1}{B(\boldsymbol{\alpha}_1)B(\boldsymbol{\alpha}_2)} \prod_{k=1}^{L} \pi_{1k}^{\alpha_{1k}-1} \pi_{2k}^{\alpha_{2k}-1} \mathrm{d}\boldsymbol{\pi}_1 \mathrm{d}\boldsymbol{\pi}_2 \\ &\quad = \frac{1}{B(\boldsymbol{\alpha}_1)B(\boldsymbol{\alpha}_2)} \sum_{j=1}^{L} \int_0^1 \sum_{h=0}^{2\rho} \binom{2\rho}{h} (-1)^h \pi_{1j}^{\frac{2\rho-h}{a}} \pi_{2j}^{\frac{h}{a}} \prod_{k=1}^{L} \pi_{1k}^{\alpha_{1k}-1} \pi_{2k}^{\alpha_{2k}-1} \mathrm{d}\boldsymbol{\pi}_1 \mathrm{d}\boldsymbol{\pi}_2 \\ &\quad = \frac{1}{B(\boldsymbol{\alpha}_1)B(\boldsymbol{\alpha}_2)} \sum_{j=1}^{L} \sum_{h=0}^{2\rho} I_j^h, \end{aligned} \quad (5.39)$$

where

$$I_j^h = \int_0^1 \binom{2\rho}{h} (-1)^h \pi_{1j}^{\frac{2\rho-h}{a}} \pi_{2j}^{\frac{h}{a}} \prod_{k=1}^{L} \pi_{1k}^{\alpha_{1k}-1} \pi_{2k}^{\alpha_{2k}-1} \mathrm{d}\boldsymbol{\pi}_1 \mathrm{d}\boldsymbol{\pi}_2. \quad (5.40)$$

Let $\hat{\boldsymbol{\alpha}}_1^{j,h}$ such as $\hat{\alpha}_{1k}^{j,h} = \alpha_{1k} + \frac{2\rho-h}{a}$, if $k = j$, and $\hat{\alpha}_{1k}^{j,h} = \alpha_{1k}$, if $k \neq j$. Take also $\hat{\boldsymbol{\alpha}}_2^{j,h}$ such as $\hat{\alpha}_{2k}^{j,h} = \alpha_{2k} + \frac{h}{a}$, if $k = j$, and $\hat{\alpha}_{2k}^{j,h} = \alpha_{2k}$, if $k \neq j$. Then,

$$\begin{aligned} I_j^h &= \binom{2\rho}{h} (-1)^h \int_0^1 \pi_{1j}^{\alpha_{1j} + \frac{2\rho-h}{a} - 1} \pi_{2j}^{\alpha_{2j} + \frac{h}{a} - 1} \prod_{\substack{k=1 \\ k \neq j}}^{L} \pi_{1k}^{\alpha_{1k}-1} \pi_{2k}^{\alpha_{2k}-1} \mathrm{d}\boldsymbol{\pi}_1 \mathrm{d}\boldsymbol{\pi}_2 \\ &= \binom{2\rho}{h} (-1)^h B(\hat{\boldsymbol{\alpha}}_1^{j,h}) B(\hat{\boldsymbol{\alpha}}_2^{j,h}). \end{aligned} \quad (5.41)$$

■

Theorem 5. *Take the Minkowski distance in a* 2τ*-norm space* ($\tau = 1, 2, \ldots$) *to define the pm-NLPs. For the CEG* $\mathbb{D}$ *presented in Definition 28 whose stages* u_1 *and* u_2 *have* L *emanating edges and* $\rho = \tau$*, then*

$$K = \sum_{j=1}^{L} \sum_{h=0}^{2\tau} \left[\binom{2\tau}{h} (-1)^h \frac{f(\alpha_{1j}, 2\tau - h) f(\alpha_{2j}, h)}{f(\bar{\alpha}_1, 2\tau - h) f(\bar{\alpha}_2, h)} \right] \quad (5.42)$$

and

$$K^* = \sum_{j=1}^{L}\sum_{h=0}^{2\tau}\left[\binom{2\tau}{h}(-1)^h \frac{f(\alpha_{1j}^*, 2\tau-h)f(\alpha_{2j}^*, h)}{f(\bar{\alpha}_1^*, 2\tau-h)f(\bar{\alpha}_2^*, h)}\right], \tag{5.43}$$

Proof. If $\nu \in \mathbb{R}_+$ and $z \in \mathbb{N}_+$, than $\Gamma(\nu+z) = \Gamma(\nu)\prod_{i=0}^{z-1}(\nu+i)$ (Abramowitz and Stegun, 1972). Now take $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \boldsymbol{a}$, where $\boldsymbol{\alpha} \in \mathbb{R}_+^n$ and $\boldsymbol{a} \in \mathbb{N}^n$. After using the previous factorisation property of gamma function and organizing the products in a convenient way, we obtain that

$$\begin{aligned} B(\hat{\boldsymbol{\alpha}}) &= \frac{\prod_{i=1}^n \Gamma(\alpha_i)}{\Gamma(\bar{\alpha})} \frac{\prod_{i=1}^n G_{a_i}\big(\prod_{j=0}^{a_i-1}(\alpha_i+j)\big)}{G_{\bar{a}}\big(\prod_{j=0}^{\bar{a}-1}(\bar{\alpha}+j)\big)} \\ &= B(\boldsymbol{\alpha}) \frac{\prod_{i=1}^n G_{a_i}\big(\prod_{j=0}^{a_i-1}(\alpha_i+j)\big)}{G_{\bar{a}}\big(\prod_{j=0}^{\bar{a}-1}(\bar{\alpha}+j)\big)}. \end{aligned} \tag{5.44}$$

After some algebra rearrangement the result follows directly from Equation 5.44 and Lemma 2 when we set the parameter $a = 1$ and $\rho = \tau$. ■

Corollary 6. *Take the Euclidean distance to define the pm-NLPs. For the CEG $\mathbb{D}$ presented in Definition 28 whose stages u_1 and u_2 have L emanating edges and $\rho = 1$, then $K = g(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ and $K^* = g(\boldsymbol{\alpha}_1^*, \boldsymbol{\alpha}_2^*)$ where*

$$g(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) = \sum_{j=1}^{L}\left[\frac{\gamma_{1j}(\gamma_{1j}+1)}{\bar{\gamma}_1(\bar{\gamma}_1+1)} - 2\frac{\gamma_{1j}\gamma_{2j}}{\bar{\gamma}_1\bar{\gamma}_2} + \frac{\gamma_{2j}(\gamma_{2j}+1)}{\bar{\gamma}_2(\bar{\gamma}_2+1)}\right]. \tag{5.45}$$

Proof. This follows directly from Theorem 5 when we set the parameter $\tau = 1$. ■

Theorem 6. *Take the distance $d(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = \|\sqrt[2\tau]{\boldsymbol{\pi}_1} - \sqrt[2\tau]{\boldsymbol{\pi}_2}\|_{2\tau}$, where $\|\,\boldsymbol{\cdot}\,\|_{2\tau}$ is the 2τ-norm ($\tau \in \mathbb{N}^+$), to define the pm-NLPs. For the CEG D presented in Definition 4 whose stages u_1 and u_2 have L emanating edges and $\rho = \tau$, then*

$$K = \frac{1}{B(\boldsymbol{\alpha}_1)B(\boldsymbol{\alpha}_2)}\sum_{j=1}^{L}\sum_{h=0}^{2\tau}\left[\binom{2\tau}{h}(-1)^h B(\hat{\boldsymbol{\alpha}}_1^{j,h})B(\hat{\boldsymbol{\alpha}}_2^{j,h})\right], \tag{5.46}$$

where

$$\hat{\alpha}_{1k}^{j,h} = \begin{cases} \alpha_{1k} + 1 - \frac{h}{2\tau} & \text{if } k = j, \\ \alpha_{1k} & \text{if } k \neq j. \end{cases} \qquad \hat{\alpha}_{2k}^{j,h} = \begin{cases} \alpha_{2k} + \frac{h}{2\tau} & \text{if } k = j, \\ \alpha_{2k} & \text{if } k \neq j. \end{cases}$$

and

$$K^* = \frac{1}{B(\boldsymbol{\alpha}_1^*)B(\boldsymbol{\alpha}_2^*)} \sum_{j=1}^{L} \sum_{h=0}^{2\tau} \left[\binom{2\tau}{h} (-1)^h B(\hat{\boldsymbol{\alpha}}_1^{*j,h}) B(\hat{\boldsymbol{\alpha}}_2^{*j,h}) \right], \tag{5.47}$$

where

$$\hat{\alpha}_{1k}^{*j,h} = \begin{cases} \alpha_{1k}^* + 1 - \frac{h}{2\tau} & \text{if } k = j, \\ \alpha_{1k}^* & \text{if } k \neq j. \end{cases} \qquad \hat{\alpha}_{2k}^{*j,h} = \begin{cases} \alpha_{2k}^* + \frac{h}{2\tau} & \text{if } k = j, \\ \alpha_{2k}^* & \text{if } k \neq j. \end{cases}$$

Proof. This follows directly from Lemma 2 when we set the parameter $a = 2\tau$ and $\rho = \tau$. ■

Corollary 7. *Take the Hellinger distance to define the pm-NLPs. For the CEG $\mathbb{D}$ presented in Definition 4 whose stages u_1 and u_2 have L emanating edges and $\rho = 1$, the normalisation constants K and K^* are given by the forms below:*

$$K = 2 - 2 \sum_{j=1}^{L} \frac{h(\alpha_{1j}, \alpha_{2j})}{h(\bar{\alpha}_1, \bar{\alpha}_2)} \tag{5.48}$$

and

$$K^* = 2 - 2 \sum_{j=1}^{L} \frac{h(\alpha_{1j}^*, \alpha_{2j}^*)}{h(\bar{\alpha}_1^*, \bar{\alpha}_2^*)}, \tag{5.49}$$

where

$$h(\gamma_1, \gamma_2) = \frac{\Gamma(\gamma_1 + 0.5)\Gamma(\gamma_2 + 0.5)}{\Gamma(\gamma_1)\Gamma(\gamma_2)}. \tag{5.50}$$

Proof. Using Equation 5.44, it follows directly from Theorem 6 when we set the parameter $\tau = 1$. ■

Thus I have shown that standard Dirichlet local priors work suboptimally when used in conjunction with the AHC algorithm. This occurs because their corresponding BF scores only take into consideration the probability masses of the stages regardless of their relative location in the probability space. Although it is important not to overstate this problem - conjugate model search is not bad - by introducing a priori a separation measure between stages NLPs tend to perform much better. Their associated BF scores corresponds to the standard local prior BF scores plus a penalisation term as function of the expected distances between stages.

However the use of product NLPs (fp-NLPs and pp-NLPs) are extremely computationally slow. Their penalisation term can also mislead the AHC algorithm since the set of stages used to define them are often bigger than the search neighbourhood of the AHC algorithm (only a pair of stages). In contrast the AHC algorithm using pm-NLPs help us efficiently identify robustly parsimonious models which conjugate or product NLPs cannot.

5.3.1 OAHC Algorithm using pm-NLPs

To incorporate efficiently the pm-NLPs into the OAHC algorithm, we should initialise distinct vectors to keep the lpBF associated with local priors and the non-local penalties $\ln K - \ln K^*$ in memory. Since the containing model has a LP and the contained model has a NLP, these two vectors avoids the need to recalculate at every loop (line 10 in Algorithm 11) all local scores. Using the lexicographic order enables us to update only the local scores corresponding to the pair of stages that are merged at each time (line 15 in Algorithm 11). Observe that the OAHC algorithm using NLPs (Algorithm 11) has an analogous algorithmic structure of the OAHC Algorithm using LPs (Algorithm 3). Thus these algorithms have the same order of computational cost in terms of memory and processing time. We now illustrate this new selection method using NLPs.

5.4 Computational Experiments

In this section I compare BF model selection with different non-local and local priors as a function of the hyper-parameter $\bar{\alpha}$ using computational simulations. These experiments enable us to study how these CEG model selection methods can explain the impact of the explanatory variables appear to have on childhood hospitalisations. I then proceed to analyse the real data set.

My second example searches over a much larger space of models. Its hypotheses concern the nature of the radicalization processes in a prison population. For reasons of confidentiality the data set I used was created through a simulation

Algorithm 11: OAHC Algorithm using Dirichlet pm-NLPs

Input: A complete data set $\boldsymbol{D}$, an event tree $\mathcal{T}$, a hyper-stage structure $\mathcal{H}$ and a parameter $\bar{\alpha}$.

Output: The best scoring CEG found.

Initialise the array U with the stage structure of $\mathbb{C}_0$ indexed by each hyper-stage $\mathcal{H}_h \in \mathcal{H}$, i.e. $|U| = |\mathcal{H}|$.

Obtain the conditional frequency tables ($\boldsymbol{y}$) for each stage of $\mathbb{C}_0$ based on $\boldsymbol{D}$ and $\mathbb{C}_0$.

Calculate the hyperparameter $\boldsymbol{\alpha}$ for each stages of $\mathbb{C}_0$ using $\boldsymbol{D}$ and $\mathbb{C}_0$ based on conservative and uniform assumptions.

Initialise an array $score$ with the log posterior probability of $\mathbb{C}_0$.

for *every partition* $\mathcal{H}_h \in \mathcal{H}$ **do**

  Initialise a vector $lpBF$: for every pair of stages $\{u_a, u_b\} \subseteq U[h]$ lexicographically ordered, calculate the $lpBF$ using Equation 4.8, where the initial model is $\mathbb{C}_0$ and the candidate model merges $u_a = \{s_a\}$ and $u_b = \{s_b\}$.

  Initialise a vector $nlpPenalty$: for every pair of stages u_a and u_b in $U[h]$ lexicographically ordered, calculate the term $\ln K - \ln K^*$.

  $nlp.lpBF \leftarrow lpBF + nlpPenalty$.

  $stop \leftarrow FALSE$

  while *stop=FALSE and* $|U[h]| > 1$ **do**

    Take the pair of stages u_a^* and u_b^* that provides the largest score $max(nlp.lpBF)$.

    if $max(nlp.lpBF) > 0$ **then**

      $score \leftarrow score + lpBF[\{u_a^*, u_b^*\}]$

      Update $U[h]$: $u_a^* \leftarrow u_{a\oplus b}^*$, where the new stage $u_{a\oplus b}^*$ merges the previous stages u_a^* and u_b^*; and eliminate the stage u_b^*.

      Update $lpBF, nlpPenalty, nlp.lpBF$: to calculate the values with respect to the new stage u_a^*; and to eliminate the values associated with stages u_b^*.

    else

      $stop \leftarrow TRUE$

return U, $score$

calibrated to be consistent with publicly available statistics associated with the UK prison population.

Here I use only the simplest possible non-local priors, the quadratic pm-NLPS ($\rho = 1$) associated with Euclidean and Hellinger distances. Although the choice of the metric might superficially look important, at least for the examples I study below the inferences appear robust to this choice. The results using these metrics are shown to be remarkably similar. I explore the CEG model space using the OAHC algorithms using local and non-local priors; see Sections 4.3.1 and 5.3.1.

5.4.1 A Health Application

Christchurch Health and Development Study Data Set

I will first revisit the data set used in Barclay et al. (2013) and Cowell and Smith (2014). Next I will use this survey to explore various features of CEG model selection in this problem. The data set used here constitutes a small part of the Christchurch Health and Development Study (CHDS) carried out at the University of Otago, New Zealand CHDS. They correspond to a 5-year longitudinal study of rates of childhood hospitalization based on cohort of 1265 children born in 1977. The children's family were interviewed at birth, at four months and once at each the following life years until the age of five year-old. The information was collected by four different ways: a structured interview with child's mother, a diary filled up by the child's mother, hospital records and practitioner notes (Fergusson et al., 1981, 1984, 1986).

For the purpose of my analyses I model the hospital admission of a child as a function of the following three explanatory discrete variables:

- family social background: a categorical variable distinguishing between high and low levels. This variable was constructed using a latent-class model based on measures of maternal educational level and age at the child's birth, child's ethnicity, family social class and whether a child entered an adoptive, a single or two parent family.

- family economic status: a categorical variable differentiating between high and low status. This variable was obtained from a latent-class model whose inputs were income, type of accommodation, standard of living and financial difficulty associated with the family of each child.
- family life events: a categorical variable indicating whether the family of a child experiences low (0 to 5 events), moderate (6 to 9 events) or high (10 or more events) number of stressful events over the 5 years. This includes events such as death, illness, unemployment and marital disharmony.

One of the diverse objectives of this study was to explore how social and economic factors associated with the stress faced by the family can affect the risk of hospitalisation during childhood. Hospital admission is then our response variable which is assumed to be binary (No, Yes) and so signals whether a child were hospitalised at least one time during his first five life years. Only hospitalisations for respiratory infections, gastroenteritides and accidents were considered. The CHDS data set available to us has a complete record of 890 children. The corresponding summary statistics are presented in Table 5.1. For a detailed description of the collection and pre-processing of this data set, see Barclay et al. (2013) and Fergusson et al. (1986) .

Hospital	**Social Status**		**Economic Situation**		**Life Events**			**Total**
Admission	Low	High	Low	High	Low	Moderate	High	
No	289	432	480	241	290	233	198	721
Yes	94	75	127	42	39	62	68	169
Total	383	507	607	283	329	295	266	890

Table 5.1: Summary Statistics of the CHDS data set for variables Social Status, Economic Situation and Life Events against Hospital Admission

Performing an exhaustive search over the CEG model space using a dynamic programming approach, Cowell and Smith (2014) discovered that the maximum a posterior (MAP) CEG is given by the variable ordering social status, economic situation, hospital admission and life events. This unfolding sequence provides

less analytical strength to study the hospitalisation rate since the variable hospital admission is not the last one. For example, if this ordering is used to causal analysis in the Pearl's sense, it asserts that there is no causal effect of life events on hospital admission. In case of explanatory analysis, it actually loses the granularity provided by life events.

Take into consideration the objective around hospitalization rates, I decided to express the hospitalisation of a child -the response (and last) variable - in terms of the following measured sequence of explanatory variables: social status, economic situation, and life events. This is the approach suggested in Barclay et al. (2013) for CEG model selection based on the MAP BN; for more detail, see Section 4.2.2. In this section the underlying event tree of all CEG models is then the same to that one depicted in Figure 5.5.

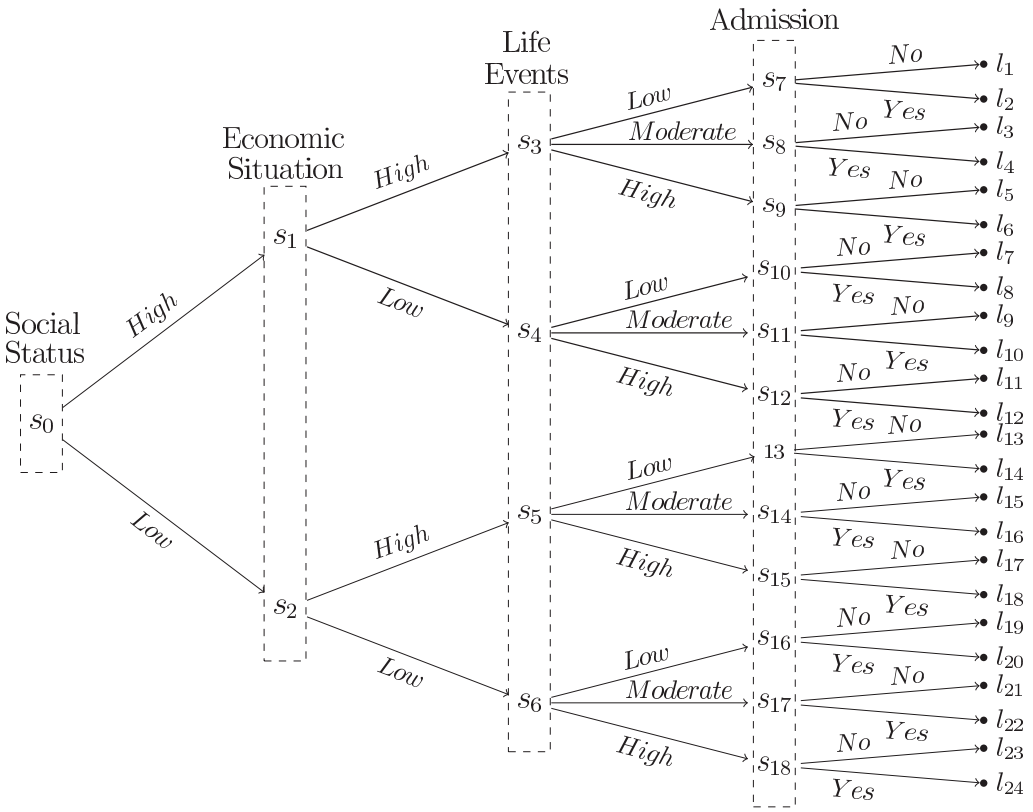


Figure 5.5: The event tree associated with the CHDS data set. Variable order: $Social\ Status \succ Economic\ Situation \succ Life\ Events \succ Admission$.

Figure 5.6 depicts the CEG model I used to generate our simulation experiments. The graphical structure corresponds to a slightly modified version of the MAP CEG found by the dynamic programming algorithm under the restriction of that variable order (Cowell and Smith, 2014); for more detail see Section 4.3.2. The conditional probabilities were assigned based on the real data set. For example, in the CHDS data set 507 enjoy high social status: 53% are in the high economic situation and 47% are in the low economic situation. So, for any unit reaching position w_1 I simulated its next development using a Bernoulli(0.47) random variable.

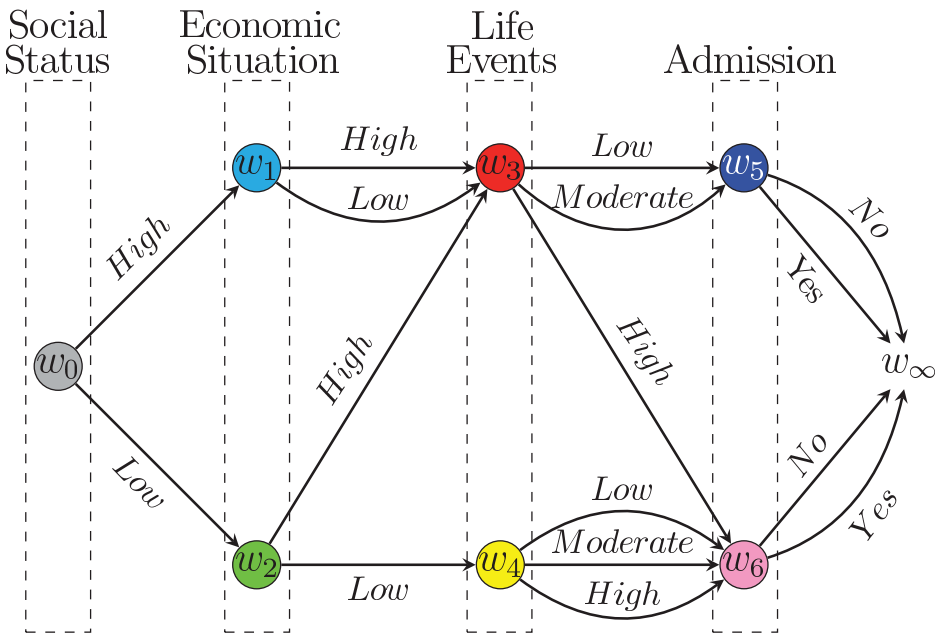


Figure 5.6: Generating CEG Model for simulation studies with the CHDS data set. Each CEG was selected using a different data set generated from the original CHDS study.

I simulated 100 samples for each sample size (SS) whose range goes from 100 to 5000 by increment of 100. For each sample, the best CEG model was selected by the OAHC algorithm for $\bar{\alpha}$-values changing from 1 to 100 by increment of 1 and also for $\bar{\alpha}$-values of 0.1, 0.25, 0.5 and 0.75. I then explored the CEG model space using both Dirichlet local priors and pm-NLPs.

Each CEG chosen was assessed using two criteria: the total number of stages,

and the total situational error. The former focus on the topological aspects of the graphical structure. For example, the generating model in Figure 5.6 has 7 stages. Its objective is to yield a summary of the graphical complexity.

The second criterion checks the overall adequacy of the conditional probabilities associated with each situation of the chosen CEG. This provides us with a diagnostic monitor to assess if the situations in the event tree are merged into stages that indeed represent the data generating model. First, define the empirical mean conditional distributional corresponding to a situation s_j, $\boldsymbol{\mu}(s_j)$, as the mean of the posterior probability distribution of the parameter $\boldsymbol{\pi}_i$ associated with the stage u_i such that $s_j \subset u_i$. Formally,

$$\boldsymbol{\mu}(s_j) = E[\boldsymbol{\pi}_i | \boldsymbol{x}, \mathbb{G}]; \qquad s_j \subset u_i. \tag{5.51}$$

The situational error $\xi(s_j)$ is the Euclidean distance between the empirical mean conditional distribution and the generating conditional distribution of a situation s_j. Thus

$$\xi(s_j) = \|\boldsymbol{\mu}(s_j) - \boldsymbol{\pi}_i^{\dagger}\|_2; \qquad s_j \subset u_i, \tag{5.52}$$

where $\boldsymbol{\pi}_i^{\dagger}$ is the conditional probability of the stage u_i in the generating model such that $s_j \subset u_i$. Finally, the total situational error $\xi(\mathcal{T})$ is obtained by the sum of situational errors over the set of situations in the event tree. We therefore have that:

$$\xi(\mathcal{T}) = \sum_{j \in \mathcal{T}} \xi(s_j). \tag{5.53}$$

To analyse the results, average values of each criterion over the 100 data sets for each pair (SS,$\bar{\alpha}$) were computed. I noted that the corresponding variance is small and does not impact the interpretation of the results presented in Figures 5.7 and 5.8. For simplicity, I have depicted below only the outcomes associated with three candidate sample sizes of 300, 900 and $3,000$. Recall that the original study was of 890 children.

Figure 5.7 shows that pm-NLPs tend to select more parsimonious CEGs than the Dirichlet local priors. Under the assumption that the CEG above is actually the true one we see that the number of stages corresponding to the CEGs chosen by

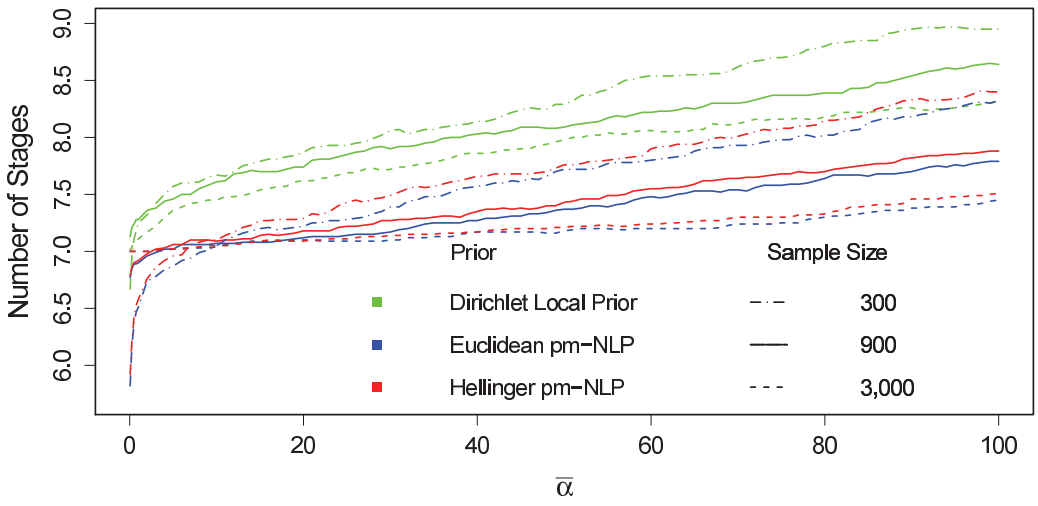


Figure 5.7: The average of the Number of Stages over the 100 CEGs selected by the OAHC algorithm according to the $\bar{\alpha}$-values 300, 900 and $3,000$ using Dirichlet LPs, Eucledian pm-NLPs and Hellinger pm-NLPs.

NLPs gets close to the true number (7) of stages over the entire range of $\bar{\alpha}$-values as the sample size increases. In contrast, the CEGs found by local priors are not greatly improved even when the sample size increases from 300 to $3,000$.

We see in Figure 5.8 that by selecting simpler graphs NLPs slightly reduce the total situational errors. This improves the CEG predictive capabilities. These errors tend to increase for larger values of the parameter $\bar{\alpha}$, particularly for small sample size. The pm-NLPs dominates the local priors consistently for a small sample size and when $\bar{\alpha}$-values are not large, and in medium and large sample sizes independently of the $\bar{\alpha}$-values. The best results appear to be concentrated around $\bar{\alpha}$-values from 1 to 20 regardless of the sample size.

The pm-NLPs appear more robust with regard to the hyper-parameter $\bar{\alpha}$. They also tend to pick more plausible models for values from 1 to 20 of this hyper-parameter regardless of the sample size. Observe that in this range the number of stages tend to be quite stable around the true number (7) and the total situational errors are minimised. On the other hand, local priors appear to give rise to substantially different inferences for different values in this parameter range. In this case although it is true that larger values of this hyper-parameter give more

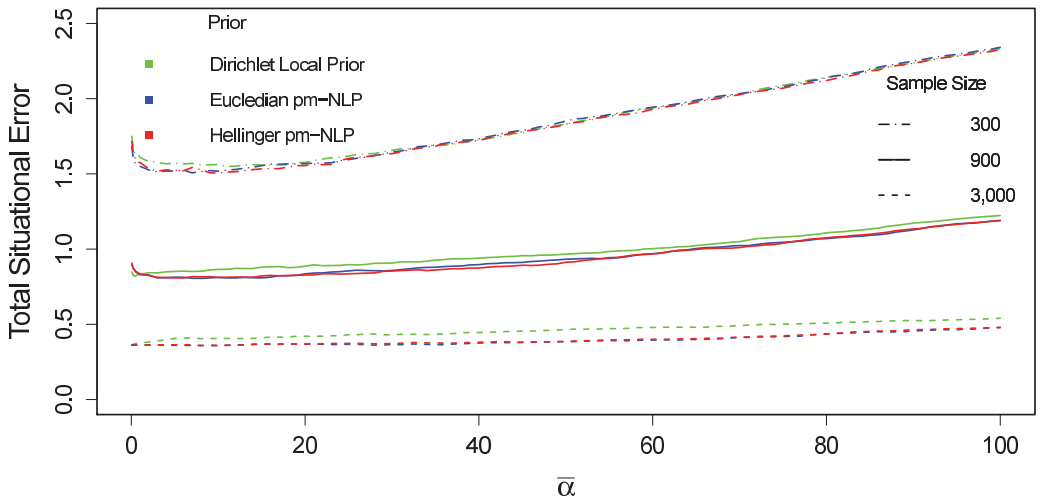


Figure 5.8: The average of the Total Situational Errors over the 100 CEGs selected by the OAHC algorithm according to $\bar{\alpha}$-values 300, 900 and 3,000 using Dirichlet LPs, Eucledian pm-NLPs and Hellinger pm-NLPs. Each CEG was selected using a different data set generated from the original CHDS study.

consistency in terms of the number of stages, these values imply larger total situational errors. They also represent very strong prior information about the various margins of individual variables: a hypothesis which would usually be a strange one to impose in many practical scenarios.

To conclude, I analyse the influence of very small $\bar{\alpha}$-values (less than 1) on the results. For a sample size of 300, although LPs tend to choose better CEGs than NLPs with regard to the number of stages, these CEGs do not optimize the total situational errors that are indeed slightly greater than those corresponding to CEGs selected by NLPs. Using the medium-size samples (900), LPs lead to more complex CEGs than the true one with respect to the number of stages whilst NLPs tend to select simpler ones, but the number of stages in both cases are the same distance from the true number (7). Here the LPs have barely smaller total situational errors than NLPs. NLPs clearly dominate the local priors in both criteria when the sample size is equal to $3,000$.

Overall very small $\bar{\alpha}$-values are not recommended since they yield very unstable results using local and non-local priors. They are also inclined to find CEGs with

larger total situational errors. In the case of pm-NLPs, these small $\bar{\alpha}$-values tend to select sparser CEG than the true one, having a strong regularization effect over the graphical structure. However the good modelling practise of calibrating a priori the predictive consequences of such prior settings would usually not encourage the choice of such values.

As expected on the basis of our theoretical results, for this example NLPs tend to be more stable and to select sparser - simpler to explain - graphs especially when compared with conventional methods. The results also indicate that NLPs are more prone to find CEGs that have a slightly better predictive capabilities for all reasonable settings of the hyper-parameter $\bar{\alpha}$.

Finally, we can observe in Figures 5.7 and 5.8 that the performances of Hellinger and Eucledian pm-NLPs are very similar in terms of the number of stages and almost identical in terms of the total situational error. However, the Eucledian pm-NLPs has a weak tendency to select CEG models with a fewer number of stages than Hellinger pm-NLPs for samples sizes smaller than $1,000$. This gives a very slight advantage for the Hellinger distance in this setting since it tends to select models that are closer to the generating model for reasonable values of the hyper-parameter $\bar{\alpha}$ (values smaller than 15). Of course, I do not believe that this fact constitutes per si sufficient evidence to prefer one metric to another.

A new analysis of the CHDS Data Set

I now compare the performance of my methods using pm-NLPs and Dirichlet local priors in a real analysis of the CHDS data set when the data generating process is assumed unknown. Figure 5.9 shows how the staged structures change as the parameter $\bar{\alpha}$ increases when I look over the CEG model space using the OAHC algorithm under the constraint of the variable order used previously.

Figure 5.9 enables us to compare the sensitivity of CEG model selection using local and using non-local priors as function of the hyper-parameter setting. Note that increasing the stability of the model selection for wider range of $\bar{\alpha}$-values

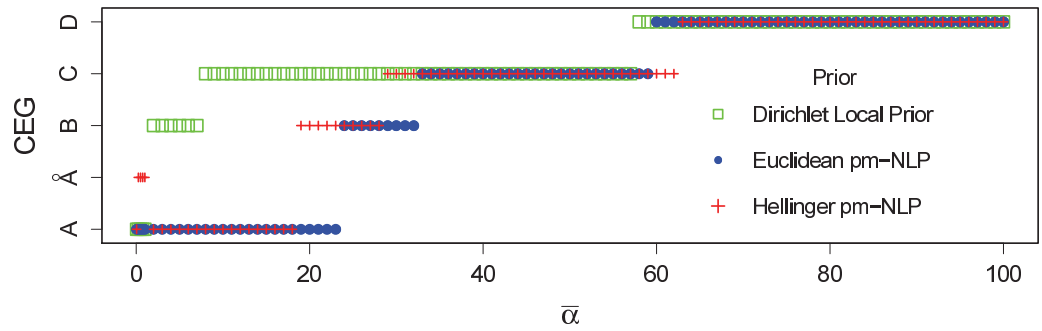


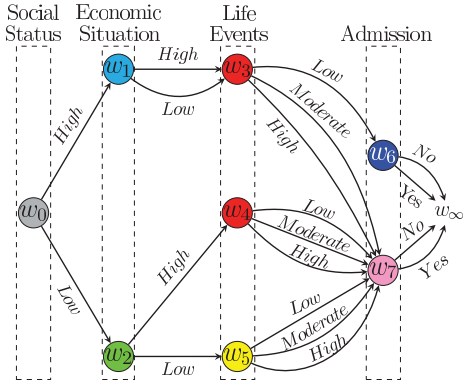
Figure 5.9: CEG Models selected by the OAHC using Dirichlet LPs, Eucledian pm-NLPs and Hellinger pm-NLPs when I set $\bar{\alpha} = 0.1, 0.25, 0.5, 0.75, 1, 2, \ldots, 100$. It was used the original CHDS data set. CEGs A, $\mathring{A}$, B, C and D are depicted in Figure 5.10.

makes the result less dependent on this hyper-parameter. The interpretation of the conditional independence statements embedded into the selected CEG then become more reliable since the choice of the CEG is unlikely to change dramaticaly with small perturbation in the $\bar{\alpha}$-values.

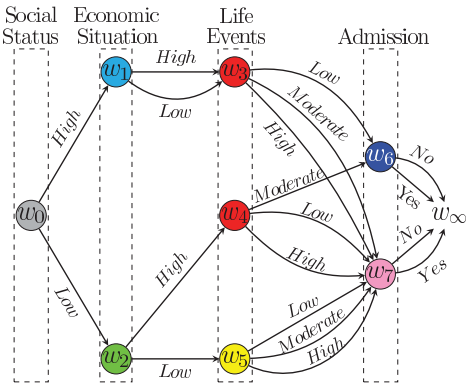
In fact, it can be seen from Figure 5.9 that local priors induce more robust results for $\bar{\alpha} \geq 8$, whilst Euclidean and Hellinger pm-NLPs are quite stable for $\bar{\alpha} \leq 23$ and $\bar{\alpha} \leq 18$, respectively. Note that the NLPs provide even more consistent outcomes of the search with regard to small and medium $\bar{\alpha}$-values: i.e. they are more robust to the setting of this hyper-parameter than the local priors. Recall from Section 5.4.1 that better results tend to be obtained by setting $1 \leq \bar{\alpha} \leq 20$.

Figure 5.10 depicts the CEG models found by the OAHC algorithm. Observe that NLPs tend to select sparser graphs. The CEGs A (Figure 5.10a) and $\mathring{A}$ (Figure 5.10b) has 7 stages, the CEG B (Figure 5.10c) has 8 stages, and the CEGs C (Figure 5.10d) and D (Figure 5.10e) have 9 stages. The OAHC algorithm using local priors points to the CEG C whilst the use of pm-NLPs indicates the CEG A. Although Hellinger pm-NLPs present some instability for very small $\bar{\alpha}$-values (1 or less), they keep pointing to the CEG A just as Eucledian pm-NLPs do. Actually the OAHC algorithm in conjunction with NLPs based on both metrics provides very closed results in general.

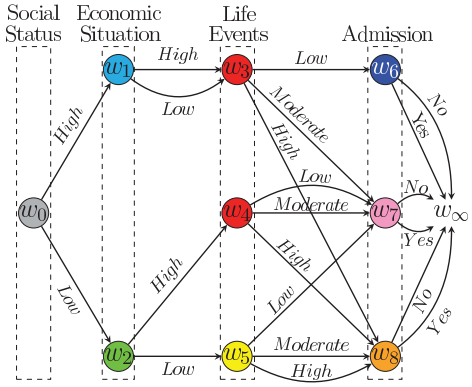
The five models selected have a tendency to be nested regardless of the type of



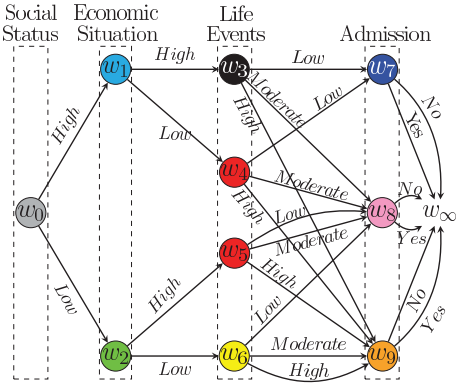
(a) CEG A



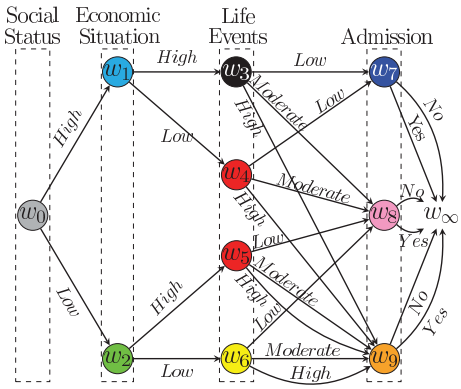
(b) CEG $\mathring{A}$



(c) CEG B



(d) CEG C



(e) CEG D

Figure 5.10: Graphical Structure of CEG Models selected by the OAHC using Dirichlet LPs, Eucledian pm-NLPs and Hellinger pm-NLPs when I set $\bar{\alpha} = 0.1, 0.25, 0.5, 0.75, 1, 2, \ldots, 100$. It was used the original CHDS data set.

priors and metric used in the NLPs. When there are some changes, they add only one or at most two stages to the CEG staged structure. For example, the CEG C is 1-nested and 2-nested in the CEGs B and A, respectively. In fact the qualitative interpretations and the probability measures do not differ very much, although the more parsimonious graphs (e.g. CEG A) give somewhat more transparent and intuitive explanations of the process. To get a better understanding of the model selection process, I chose these three models A, B and C for a more detailed analysis since they were the most frequent models selected for $\bar{\alpha} \leq 60$ regardless of priors used.

Thus observe that the CEG B is identical to the CEG C except that the variable life events has two stages (u_3, u_4) and three positions (w_3, w_4, w_5) in the CEG B, and three stages (u_3, u_4, u_5) and four positions (w_3, w_4, w_5, w_6) in the CEG C. As highlighted in red (Table 5.2), only the conditional probabilities associated with these positions have changed, and then only very slightly. Furthermore although these CEGs differ, their causal hypotheses associated with childhood hospitalisation are in fact identical: the hospital admissions are partitioned into the same three groups of patients in both.

Highlighting only the substantial differences implied by the data set, the CEG A brings new and much simplified hypotheses about how hospital admissions relate to the covariates. It proposes the existence of only two distinct risk groups of hospital admission. The CEGs B and C segment the higher risk individuals in the CEG A (position w_7) into two groups (positions w_7 and w_8). Note that the differences in the probability of hospital admission between these two groups (Table 5.2, in blue) are small. In other words, both groups continue to identify a higher risk population in comparison with individuals who experience a low number of life events and have higher social status.

Stage	Variable	State Space	Posterior Mean (%)		
			CEG A	CEG B	CEG C
u_0	Social	(h,l)	(57,43)	(57,43)	(57,43)
u_1	Economic	(h,l)	(47,53)	(47,53)	(47,53)
u_2	Economic	(h,l)	(12,88)	(12,88)	(13,87)
u_3	Life Events	(l,m,h)	(46,34,20)	(46,34,20)	(43,33,24)
u_4	Life Events	(l,m,h)	(22,31,47)	(22,31,47)	(50,36,14)
$-/u_5$	Life Events	(l,m,h)	-	-	(22,31,47)
u_5/u_6	Hospitalisation	(n,y)	(91,9)	(91,9)	(91,9)
u_6/u_7	Hospitalisation	(n,y)	(77,23)	(82,18)	(82,18)
u_7/u_8	Hospitalisation	(n,y)	-	(73,27)	(73,27)

Legend: l- Low; m- Moderate; h- High; n- No; y- Yes
$\cdot$ / $\cdot$ - CEGs A & B / CEG C

Table 5.2: Posterior mean corresponding to the stages in CEG Models A, B and C depicted in Figure 5.10 when I set $\bar{\alpha} = 3$ for CEG A, $\bar{\alpha} = 6$ for CEG B and $\bar{\alpha} = 12$ for CEG C. It was used the original CHDS data set. The stage structures of these models are given by:

- Model A: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3, w_4\}$, $u_4 = \{w_5\}$, $u_5 = \{w_6\}$, $u_6 = \{w_7\}$.
- Model B: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3, w_4\}$, $u_4 = \{w_5\}$, $u_5 = \{w_6\}$, $u_6 = \{w_7\}$, $u_7 = \{w_8\}$.
- Model C: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3\}$, $u_4 = \{w_4, w_5\}$, $u_5 = \{w_6\}$, $u_6 = \{w_7\}$, $u_7 = \{w_8\}$, $u_8 = \{w_9\}$.

5.4.2 A Security Application

Introduction

My second CEG search was conducted over a much larger class of hypotheses this time about the nature of the process of radicalisation within prisons. My main focus here is to develop methods to identify groups of individuals who are most likely to engage in specific criminal organization in British prisons. As I will show, this example is very challenging because the classes of each variable are remarkably unbalanced and the percentage of radical prisoners - those units of special interest - is tiny. Furthermore, if expressed in terms of a BN (see Figure 5.11) any plausible generating model would need to be highly context-specific: generic BN model selection methods could therefore not be expected to work well. To accommodate all different types of context-specific dependencies involved in prison radicalisation process a more flexible family such as the CEG class really does need to be used.

For the purposes of this illustration I have restricted our analyses to consider only six explanatory variables. These have been chosen because they are often hypothesised as playing a key role in the process of radicalisation. These are:

- Gender - a binary variable distinguishing between male (M) and female (F);
- Religion - a nominal variable with three categories: Rel- religious prisoner, NRel- non-religious prisoner and NRec- not recorded;
- Age - an ordinal variable with three categories: A1- age < 30, A2- $30 \leq$ age < 40 and A3- age ≥ 40;
- Offence - a nominal variable with five categories: VAP- violence against person, RBT- robbery, burglary or theft, D- drug, SO- sexual offence and O- others;
- Nationality - a binary variable differentiating between British citizens (B) and foreigners (Fo);
- Network - an ordinal variable differentiating groups of prisoners according to their social interactions with well-known members of the target criminal organisation. It has three categories: I- intense; Fr- frequent; and S- sporadic.

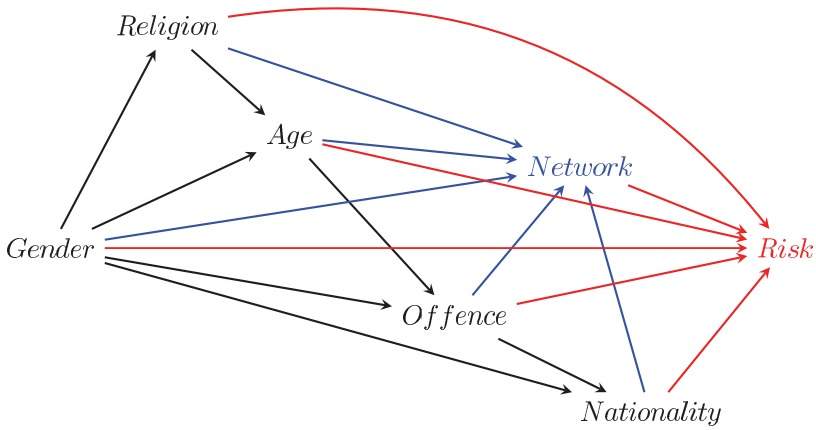


Figure 5.11: Generating BN Model for simulation studies about radicalisation within prisons.

Because of the sensitive nature of data in this field, I have based this example on a data set some of whose variables have been simulated. However I have chosen simulations that are calibrated to real figures and real hypotheses currently in the public domain concerning the British prison population (Ministry of Justice (2013)). So the simulations plausibly parallel the likely current scenario. The generating model used was based on an initially elicited BN depicted in Figure 5.11. The real data set enables us to naively estimate the joint distributions for the first five explanatory variables. These are presented in black in this figure. An important point is that several variables have sparse cell counts: for example Gender (F-5%), Religion (NRec-2%) and Nationality (F-10%).

No data was publicly available for the explanatory variable Network and the response variable Radicalisation. So in this study I instead construct a probability model over certain developments based on expert judgements (Cuthbertson (2004), Jordan and Horsburgh (2006), Hannah et al. (2008), Neumann (2010), Silke (2011), Rowe (2014)).

To perform the necessary data simulation I needed to specify the 180 conditional probability distributions of variable Network given the first five explanatory variables. Here I assumed that there are only four different social interaction mechanisms; see Table 5.3. For example, male, foreign, younger and non-religious (or not recorded) prisoners who are in jail for violence against person, robbery, burglary,

theft or drug offences are hypothesised to have the strongest tendency to become closer to individuals of the target criminal organisation.

Variable	**Generating Mechanism**	**Conditional Probability** (%) (I,Fr,S)/(H,L)*	**Number of Partitions**
Network	N1	(75,15,10)	6
	N2	(45,30,25)	23
	N3	(10,40,50)	79
	N4	(1,10,89)	72
Radicalisation	R1	(30,70)	6
	R2	(3,97)	114
	R3	(0.1,99.9)	420

*(I,Fr,S) and (H,l) are, respectively, the category vectors of Network and Radicalisation.

Table 5.3 Generating mechanisms assumed for variables Social Network and Radicalisation.

The response variable - introduced last - distinguishes between individuals at high or low risk of radicalisation. Being the last variable to be sampled for each prisoner, this has 540 conditioning partitions. In this environment risk assessments are generally coarse. So based on the expert judgements cited above these partitions are clustered into only three different radicalisation classes of risk (Table 5.3). The highest risk prisoners come from only six partitions that corresponds to those prisoners who are socially more closed to members of the target criminal organisation. Note that from a technical viewpoint these plausible hypotheses introduce several prior context-specific conditional assessments into our model.

The radicalisation risk of the whole prison population is hypothesised to be small in line with the expert judgement and academic literature (Cuthbertson (2004), Jordan and Horsburgh (2006), Hannah et al. (2008), Neumann (2010), Silke (2011)). Here this is set at around 0.7% of the total population. Based on the premises discussed above, I then simulated 100 complete data sets. Each of these

has 85,000 individuals, approximating the recent yearly totals of the British prison population. Assuming my fixed generating model is true I will now investigate the efficacy of various CEG search methods to identify those prisoners most likely to be radicalised in each of these data sets.

CEG Model Searches

Assume that our optimal model is consistent with a variable sequence Gender, Religion, Age, Offence, Nationality, Network and Radicalisation. This simplifies the search space and matches the goals of this work. The CEG model search was performed using a setting of the hyper-parameter $\bar{\alpha} = 5$. This corresponds to the maximum number of categories taken by a variable in the problem. This value is also in line with my previous results that suggest that the selection of a hyper-parameter in this region will provide robust results; see above the CHDS simulation example in Section 5.4.1. I note that this was actually confirmed numerically in additional exploratory studies within this example.

The scale of this problem requires us to use a heuristic algorithm like OAHC since the SCEG space contains more than 10^{1105} SCEG models even given the chosen variable order. Here full model search strategies such as ones using Dynamic Programming will obviously be infeasible.

As expected the results in Table 5.4 indicate that the OAHC algorithm in conjunction with pm-NLPs was prone to select more parsimonious and user-friendly models than those obtained using standard local priors especially for stages near the leaves of the corresponding event tree. The Hellinger pm-NLPs tend to find a slightly simpler models than the Euclidean pm-NLPs in terms of staged complexity. NLPs also ensured that the OAHC algorithm selected models with a number of stages associated with the variables Network and Radicalisation closer to the generating model than those achieved using the Dirichlet local priors.

The use of pm-NLPs enabled the OAHC algorithm to find CEG models that clearly better represented the simulated generating process of radicalisation. For example,

Variable	Number of Stages			Number of	Maximum
	DLP	Euc-NLP	Hel-NLP	Generating Stages	Number of Stages
Gender	1.0	1.0	1.0	1	1
Religion	2.0	2.0	2.0	≤ 2	2
Age	4.8	4.1	4.1	≤ 6	6
Offence	6.0	5.9	5.9	≤ 6	18
Nationality	7.4	5.4	5.1	≤ 10	90
Network	10.2	7.2	6.8	4	180
Radicalisation	7.6	5.6	5.3	3	540

Table 5.4: Average of the Numbers of Stages in 100 Radicalisation CEGs selected by the OAHC algorithm using Direchlet Local Priors (DLP), Eucledian pm-NLPs (Euc-NLP) and Hellinger pm-NLPs (Hel-NLP). It was generated 100 different data set.

Eucledian and Hellinger pm-NLPs classified the highest risk population spuriously in only 29 and 28 date sets, respectively, whilst local priors had problems with 39 data sets. So local priors misclassified some of the highest risk individuals in more than 34% and 39% of the data sets than Euclidean pm-NLPs and Hellinger pm-NLPs, respectively. These misclassifications using local priors and pm-NLPs were associated with the highest risk groups whose sample sizes were less than 25 and whose sample proportions of radical prisoners were concentrated around 12%. Furthermore inference using local priors struggled to identify the risk level for a high risk group of 209 individuals where the sample proportion of radical prisoners was 24%.

There were only three levels of risk of radicalisation in the generating model. So for the sake of simplicity the stages that were found by the OAHC algorithm were amalgamated in Table 5.5 according to their corresponding radicalisation risk in five categories. I matched the risks greater than 25%, between 1% and 7% and less than 1% as corresponding to the risk of 30%, 3% and 0.1% in the generating model, respectively.

Although local and non-local priors yield broadly equivalent estimates for the lower two levels of radicalisation risk, Dirichlet local priors lost track of 9 of the highly hazardous individuals on average whilst pm-NLPs only lost about 6. This means an improvement of 33% in favour of pm-NLPs. The Hellinger pm-NLPs are also a little less prone to misclassified high risk prisoners than the Euclidean pm-NLPs.

Note also that local priors unlike the pm-NLPs tend to introduce a stage at risk level between 15% and 25%. If we merged the three higher levels of radicalisation risk into one category, we would lose 3 high risk individuals on average regardless of the type of prior used. However in this case local priors would include 50 more medium risk individuals (3%) in the high category. This would correspond to almost 70% more prisoners that as a result of the analysis would be spuriously identified as a danger to the public.

Although the model used here is rather naive and our results are not perfect, this larger example does nevertheless demonstrate the promise of pm-NLPs used in conjunction with a greedy search of CEG models when applied to much larger scale asymmetric populations like the one above.

		Dirichlet Local Prior - Errors					Eucledian pm-NLP - Errors					Hellinger pm-NLP - Errors					Number of
SCEG Risk(%)		≥ 25	(15,25]	(7,15]	(1,7]	≤ 1	≥ 25	(15,25]	(7,15]	(1,7]	≤ 1	≥ 25	(15,25]	(7,15]	(1,7]	≤ 1	Prisoners
Generating Model Risk	30	-8.9	2.5	3.1	3.0	0.3	-5.5	0	1.6	3.6	0.3	-4.9	0	1.2	3.6	0.3	699
	3	16.4	0.2	111	-887	759	19.4	0	57	-844	768	17.8	0	66.3	-853	769	119×10^2
	0.1	0.9	0	3.5	359	-363	1.1	0	1.4	373	-375	1.1	0	1.0	330	-333	724×10^2

Table 5.5: Average Number of misclassified prisoners in 100 CEGs selected by the AHC algorithm according to their risk of radicalisation in the Generating Model

Chapter 6

A Dynamic Chain Event Graph

A Dynamic CEG (DCEG) (Barclay et al., 2015) and a CEG (Smith and Anderson, 2008) are both graphical models obtained after eliciting an event tree and embellishing it into a staged tree. The main difference between these two class of models is that a DCEG is based on an infinite tree whilst a CEG is supported by a finite tree. Therefore, a strong connection exists between the two classes and the topological semantic of both models is closely related. It is therefore natural and sometimes easier to explore some ideas using first a CEG up to a certain time and only afterwards to then extend these to a DCEG.

In Barclay et al. (2015) we directly extended the CEG semantics based on a finite tree to an infinite one in order to define a very general DCEG. Doing this we showed that it is possible to obtain a finite DCEG graph and use the Dirichlet-multinomial framework to learn the dynamic model. Despite being a significant advance in this area, there are some technical challenges that needed to be addressed. For instance, being based on an infinite tree a DCEG model demands a flexible and efficient framework to represent our target process. Otherwise, it will not be possible to depict the event tree graphically and to encode it for computational processing. This requires us to encapsulate finite sub-processes in structures that enable us to construct a large-scale model for the whole infinite process based on a finite set of sub-processes using a hierarchical modelling approach.

Another challenge is that the definition of a DCEG (Barclay et al., 2015) may lead us to some topological inconsistencies. This type of problems arises because the current concept of position does not enforce a bijective map between the infinite staged tree and its corresponding DCEG graph. Therefore, the definition of position does not always preserve the time-slice structure. This is important because to obtain a finite DCEG graph it is necessary to propose a graphical semantic that enforces a loop over time-slices and not within a time-slice. As we are working with an infinite tree if we do not add some further structure to the idea of position we can end with a loop within a time-slice. In this case, the readers of a DCEG model cannot determine when a unit will get out of a loop. Example 8 illustrates this drawback. Note that if a process can also be represented by context-specific DCEG models, this kind of problems will often not happen. The main problem in Barclay et al. (2015) is that we naively translated the concepts from a CEG to a DCEG without formally defining the probabilistic model and rigorously examining the results.

Example 8. Take two discrete random variables X and Y. The variable X may assume values 1 and 2. The variable Y takes on one of the values l or h. Now consider the following two processes that units may follow in a system.

- Process 1
  - at time-slice t=0
    - it may happen events a or b.
  - at time-slice t=1
    - if event a happens at $t = 0$, then it will be observed variable X. If variable $X = 1$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.
    - if event b happens at $t = 0$, then it will be observed variable Y. If variable $Y = h$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.
  - at time-slice $t = N, N \geq 2$

* if the previous event is $X = 1$, then it will be observed variable Y. If variable $Y = h$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.
* if the previous event is $Y = h$, then it will be observed variable X. If variable $X = 1$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.

- Process 2
  - at time-slice t=0
    * it may happen events a or b.
  - at time-slice t=1
    * if event a happens at $t = 0$, then it will be observed variable X. If variable $X = 1$, then it will be observed variable Y, otherwise a unit will get out of the system. If variable $Y = h$, then it will be observed variable X again, otherwise a unit will get out of the system. Finally, if variable $X = 1$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.
    * if event b happens at $t = 0$, then it will be observed variable Y. If variable $Y = h$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.
  - at time-slice $t = N, N \geq 2$
    * if the last event at $t = N - 1$ is $X = 1$, then it will be observed variable Y. If variable $Y = h$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.
    * if the last event at $t = N - 1$ is $Y = h$, then it will be observed variable X. If variable $X = 1$, then it will be observed variable Y, otherwise a unit will get out of the system. If variable $Y = h$, then it will be observed variable X again, otherwise a unit will get out of the system. Finally, if variable $X = 1$, then a unit will go to the next time-slice, otherwise a unit will get out of the system.

These are different processes. In the first process it happens only one event during each time-slice. In the second process it may happen one, two or three events during each time slice t, $t = 1, 2, \ldots$. However we obtain the same DCEG graph (Figure 6.1) for both processes if we use the definition of position as described in Barclay et al. (2015). This occurs because there are loops over events that happen at different time-slices in the first process whilst there are loops over events that may happen within the same time-slice in the second process.

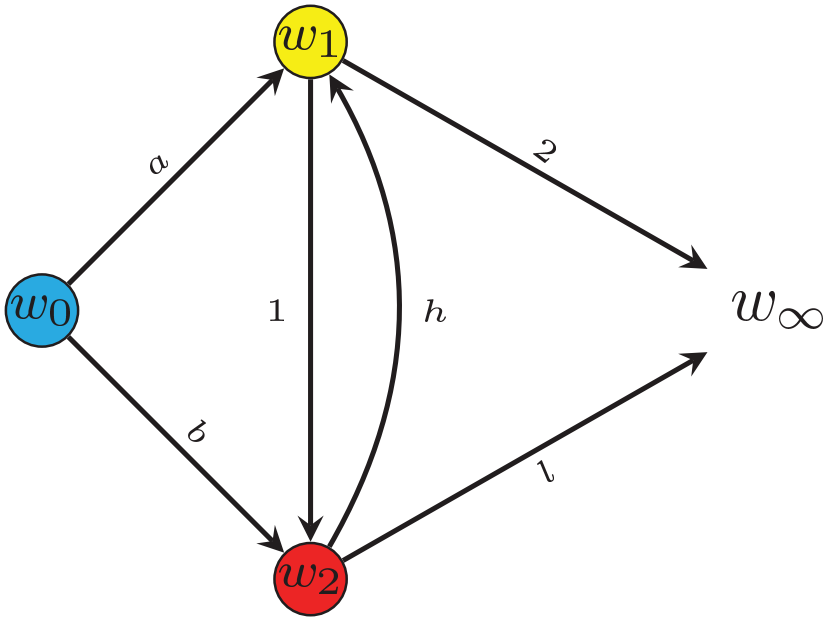


Figure 6.1: DCEG associated with the two processes described in Example 8. This graph follows the DCEG semantics proposed in Barclay et al. (2015).

□

In this chapter I address these challenges. First, to make the class of DCEG models widely applicable it is vital to develop a systematic framework to elicit an infinite tree in a compact and useful way. Second, the σ-algebra associated with a DCEG model has to be carefully defined. This can then enable us to identify and introduce new topological and probabilistic objects based on an infinite tree. Third, the definition of positions has to be reviewed. In Barclay et al. (2015) we used the same CEG concept over an infinite tree to obtain a DCEG. However except in very well-behaved and highly symmetric models this definition is overly restrictive and incapable of expressing topologically some common collections of

hypotheses we may well want to express. Finally, the necessary and sufficient conditions to construct a finite DCEG from an infinite probabilistic tree is also an open question.

The material in this chapter constitutes an *entirely original contribution* to the field. I will first present the topological concepts associated with the first two of these steps taken to obtain a DCEG: the elicitation of an event tree and its subsequent embellishment into a staged tree. This motivates me to propose a new way of constructing an event tree based on process-driven objects. I will then be able to use these objects to formally define the probability space associated with a given DCEG.

I will also propose a broader definition of a staged tree. This will enable us to increase the expressiveness of a DCEG model without losing any of its desirable properties. I argue that by adding time-invariant variables we are able to extend this framework in a simple and transparent way. This provides a very flexible family of graphical models capable of addressing a wide range of discrete dynamic processes. I then proceed to formally define a DCEG. A formal characterisation of a finite DCEG is given in terms of graphical periodicity and time-homogeneity.

6.1 Modelling a process using an Event Tree

By focusing on the qualitative description of a process, an event tree is an important tool because it provides a framework around which a technical expert can explain the process that needs to be statistically modelled. However an event tree may quickly become large, both in width and depth. To keep it tractable and useful in practice, it is therefore important to develop a systematic framework enabling us to represent it more compactly. To obtain some insights about how to do it, consider first the example below.

Example 2 (Radicalisation Process - cont.). Recall the statistic radicalisation process described in Section 2.3. The event tree in Figure 6.2 provides a snapshot of the multiple ways that this process (Example 2) can unfold for each inmate.

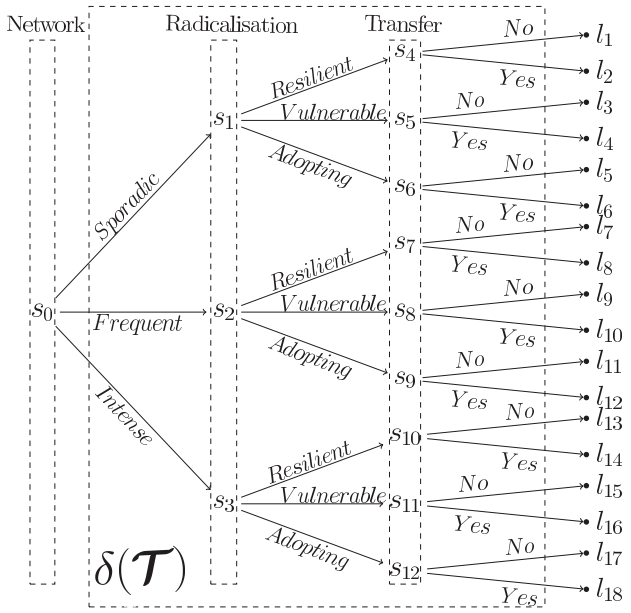


Figure 6.2: Finite Event Tree associated with the radicalisation process described in Example 2.

Note that the multiple states of this process can be graphically hidden using the big rectangle $\delta(\mathcal{T})$ in Figure 6.2 that encompasses all situations between the root vertex and the set of leaf vertices. In doing this we are able to represent an event tree as a special tree object $\Delta(\mathcal{T})$. This emphasises only the starting situation of the process (situation s_0) and its possible outcomes (leaves l_i, $i = 1, \ldots, 18$). To illustrate this convention the event tree of Figure 6.2 is schematically depicted by the tree object *$\Delta(\mathcal{T})$ in Figure 6.3. Note that in this case the rectangular vertex summarily represents a forest graph whose components are the florets associated with positions s_1, s_2 and s_3.* □

Let $l(\mathcal{T})$ be the set of leaf nodes of an event tree $\mathcal{T}$ and $V(\Delta)$ be the interface set constituted by the root and leaf nodes of $\mathcal{T}$. Drawing some analogies with Object-Oriented BNs (Koller and Pfeffer, 1997, Bangsø and Wuillemin, 2000) - for a short presentation, see Section 2.6 - in terms of encapsulating information using objects that can be connected among themselves only via interface nodes I now formally define a tree object as in Definition 29.

Definition 29 (Finite Tree Object)**.** A *finite tree object* $\Delta(\mathcal{T})$ is a graphical object that compactly depicts a finite event tree $\mathcal{T}$ by a rectangular vertex $\delta(\mathcal{T})$, the point

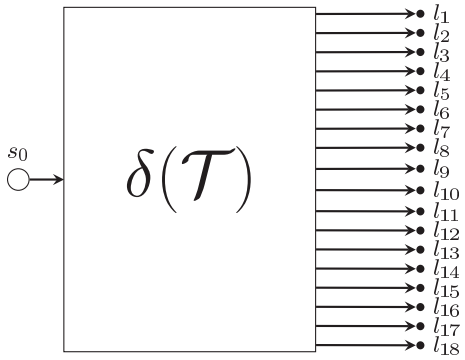


Figure 6.3: The *tree object* $\Delta(\mathcal{T})$ that symbolises the event tree depicted in Figure 6.2.

vertex set $V(\Delta)$ and a set of direct edges $E(\Delta) = \{(s_0, \delta)\} \cup \{(\delta, s); s \in l(\mathcal{T})\}$, such that:

1. the initial node $s_0, s_0 \in \mathcal{T}$, is upstream of the rectangular vertex $\delta(\mathcal{T})$;
2. all leaf nodes in $l(\mathcal{T})$ are downstream of the rectangular vertex $\delta(\mathcal{T})$; and
3. the rectangular vertex $\delta(\mathcal{T})$ represents a forest graph whose components are the event subtrees of $\mathcal{T}(s_i)$ that unfold from each situation s_i, $s_i \in ch(s_0)$, until reaching a subset of situations in $\{s_j; s_j \in pa(l_i)$ for some $l_i \in l(\mathcal{T})\}$. If $\mathcal{T}$ is a star graph then the rectangular vertex $\delta(\mathcal{T})$ represents an empty graph.

A tree object is a process-driven rectangular object embellished with root and leaf nodes. Analogous to a compact graphical representation of a BN object (Bangsø and Wuillemin, 2000), it hides the unfolding of the process and keeps visible only the interface set $V(\Delta)$ which represents the initial state and the possible final states of the process. The interface set enables us to combine together sub-processes that are components of an ongoing process.

Doing this we can link the same tree object $\Delta(\mathcal{T})$ with different leaf vertices l'_i of a given event tree $\mathcal{T}'$. In this case, the node s_0 in $\Delta(\mathcal{T})$ plays the role of an input vertex that assumes the value corresponding to the state of a particular unit at a leaf $l'_i \in l(\mathcal{T}')$. The leaf nodes l_i in $\Delta(\mathcal{T})$ are the set of output states that a unit arriving at l'_i can be after developing according to the local process depicted by $\Delta(\mathcal{T})$, i.e. the logical concatenation of event propositions representing by the

states l_i' and l_i. In this sense, the rectangular vertex $\delta(\mathcal{T})$ can be interpreted as a symbolic transformation of states associated with a particular local process.

To model a longitudinal process, a useful strategy is to just explore particular types of domain information over time. It is straightforward to capture this structure by constructing a DCEG using process-driven objects defined according to the time-slices and the unfolding paths of the event tree. The idea is to construct subtrees corresponding to a subprocess at time-slice t given a particular path in the event subtree that describes the development up to the current time point. Using such a representation, parallel panels of experts can take part in the construction of the whole event tree that can then integrate the domain beliefs coherently and be compactly presented in blocks. For technical consistency, henceforth I assume that only a finite number of events may happen over each time-slice associated with the development of a process.

So, take a situation s_i at any time-slice before the beginning of interval $t+1$ and let $\mathcal{T}_t(s_i)$ denote the finite tree that unfolds from s_i and stops at the end of interval t. Now write $\mathcal{T}_t \equiv \mathcal{T}_t(s_0)$ and let $\mathcal{T}(s_i)$ be the whole event tree that unfolds from s_i. When $\mathcal{T}(s_i)$ is an infinite tree, I sometimes write $\mathcal{T}_\infty(s_i)$ to highlight this fact. The unfolding process in time-slice t is then represented by a collection of event subtrees $\mathcal{F}o_t = \{\mathcal{T}_t(s_i); s_i \in l(\mathcal{T}_{t-1})\}$. This collection then constitutes a forest graph whose components are given by $\mathcal{T}_t(s_i)$. I also use the convention that in an event tree a terminating event is indicated by a diamond shape vertex. This framework is illustrated in the example below.

Example 3 (Dynamic Radicalisation Process - cont.). Figure 6.4 shows the infinite event tree associated with Example 3, where the symbol "..." represents implicit continuations of the infinite tree. Note that the process terminates at situations $s_{14}, s_{22}, s_{30}, s_{60}, s_{300}$ *and* s_{420} *because a prisoner transfers. Also observe that in this particular example, although this is not a necessary feature in general, every process-driven object happens to be topologically identical to the one depicted by the finite tree* $\mathcal{T}$ *in Figure 6.2.*

The finite tree $\mathcal{T}_1(s_{13})$ in Figure 6.4 above summarises what can happen at time-slice 1 to a prisoner who keeps sporadic social contacts with identified extremist recruiters, is resilient to the radicalisation process and has been not transferred at the initial interval. The event tree $\mathcal{T}_1$ depicts the whole set of events that can unfold from the initial situation s_0 to the end of time-slice 1.

The infinite event tree $\mathcal{T}_\infty(s_{13})$ rooted at situation s_{13} describes all possible events that can happen to a prisoner who has not been transferred and has interacted sporadically with extremist ideologists but has not presented a tendency to be radicalised within the initial time-slice. The forest

$$\mathcal{F}o_1 = \{\mathcal{T}_1(s_i); i = 13, 15, \ldots, 29\} \cup \emptyset$$

then depicts how radicalisation, transfer and network events associated with a prisoner can unfold at time-slice 1. Observe that the empty set is included in $\mathcal{F}o_1$ to stress that the process terminates at situation $s_i, i = 14, 16, \ldots, 30$.

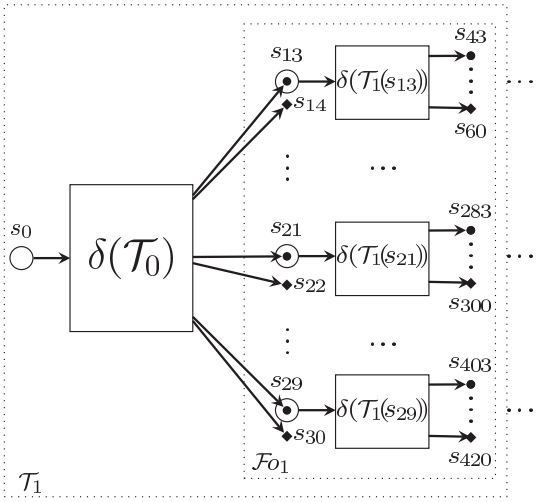


Figure 6.4: The representation of the infinite Event Tree associated with Example 3 using tree objects. A dotted rectangle establishes the limit of a particular graph.

□

As is the case when building blocks elicited during the construction of an object-oriented BNs (Bangsø and Wuillemin, 2000, Neil et al., 2000), *tree* objects can also be defined using different domain aspects other than the overarching time-slice division. This property can be especially useful for incorporating time-invariant covariates into our models. In this case, we first need to elicit an event tree

$\mathcal{T}_{-1}$ associated with this set of covariates. Only afterwards do we plug-in the tree objects corresponding to processes that unfold from each leaf of $\mathcal{T}_{-1}$. Note that each root-to-leaf path in $\mathcal{T}_{-1}$ characterises a particular type of units that is observed in a system. Example 9 illustrates this construction. Henceforward, let $\mathcal{T}_{-1}$ be an event tree associated with a set of time-invariant covariates.

Example 9 (Extended Dynamic Radicalisation Process). *Suppose that Example 3 refers to a British prison. Assume that all previous conditions continue to hold. We would like now to control the radicalisation and transfer processes in this prison for prisoners' previous conviction (Yes or No) and nationality (British or Foreign). To represent this dynamic using an event tree is straightforward as showed in Figure 6.5.*

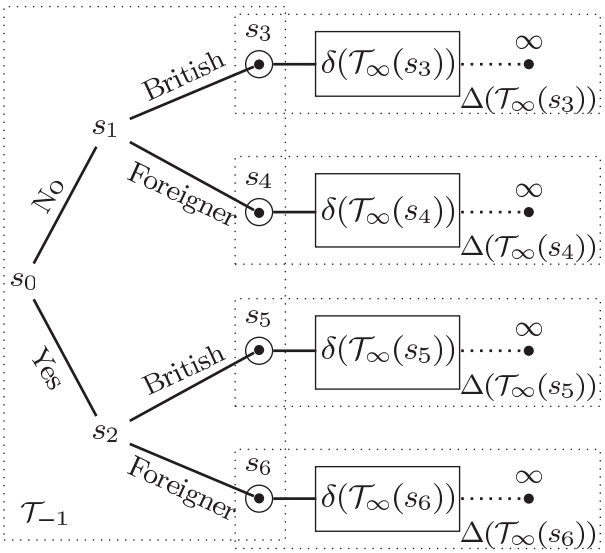


Figure 6.5: The infinite Event Tree associated with the dynamic radicalisation process controlled by two time-invariant variables: Previous Convection and Nationality. The situation s_0 corresponds to the variable Previous Conviction and the situations s_1 and s_2 are associated with the variable Nationality. The objects $\Delta(\mathcal{T}_\infty(s_i))$, $i = 3, \ldots, 6$, represent identical infinite event trees. A dotted rectangle establishes the limit of a particular graph.

Note that the leaves of $\mathcal{T}_{-1}$ – $l(\mathcal{T}_{-1}) = \{s_i, i = 3, \ldots, 6\}$– characterise four type of inmates: s_1 – British non-convicted inmates; s_2 – Foreigner non-convicted inmates; s_3 – British convicted inmates; and s_4 – Foreigner convicted inmate.

The driven-process objects $\Delta(\mathcal{T}_\infty(s_i)), i = 3, 4, 5, 6$, represent infinite event trees. These infinite tree objects differentiate from those defined for finite trees in two aspects. First, the leaf vertices of a finite tree are replaced by a single vertex labelled by a symbol ∞. Second, this vertex ∞ is connected to the rectangular vertex $\delta(\mathcal{T}_\infty(s_i))$ by a dashed line. It is assumed here that these infinite trees are identical to the one given in Figure 6.4 although this assumption could be relaxed.

In more complex real-world scenarios, this framework also enables us to conduct a distributed model construction that composes coherently the domain information. For instance, we could organise distinct teams that would be in charge of modelling the corresponding processes associated with prisoners who has previous criminal charges or not. Subsequently the results could be unified using these driven-process objects that could be further split to allow the identification of finer commonalities between the processes ($\mathcal{T}_\infty(s_1)$ and $\mathcal{T}_\infty(s_2)$) in these two prison sub-populations.

□

Although some care is needed it is nevertheless important to formalise mathematically the object-recursive approach developed above. So, take a finite event tree $\mathcal{T}$ and denote by l_i a situation s_i represented by a leaf vertex in $\mathcal{T}$. Let $\Upsilon(\mathcal{T}) = \{\Upsilon_k; k = 1, \ldots, n\}$, where $\Upsilon_k = \{l_{k_i}; i = 1, \ldots, n_k\}$ is a partition of the leaf vertex set of $\mathcal{T}$. For example, using the tree in Figure 6.4 we can choose

$$\Upsilon(\mathcal{T}) = \{\Upsilon_1 = \{s_{13}, s_{14}, \ldots, s_{18}\}, \Upsilon_2 = \{s_{19}, s_{21}, \ldots, s_{29}\}, \Upsilon_3 = \{s_{20}, s_{22}, \ldots, s_{30}\}\}.$$

Note that each partition identifies prisoners according to the particular set of unfolding events. So for example Υ_1 characterizes prisoners who have few social contacts with extremist recruiters, whilst Υ_3 and Υ_2 are defined by prisoners with frequent or intense social contact with extremist ideologists and, respectively, were transferred or not. I proceed to define the merging operation $\mathcal{T} \uplus_h \Gamma$ between a finite tree $\mathcal{T}$ and a set of trees $\Gamma = \{\mathcal{T}_i\}$ according to a map $h : \Upsilon(\mathcal{T}) \to \Gamma$.

Definition 30 (Merging Operation)**.** The *merging operation* $\boldsymbol{\mathcal{T} \uplus_h \Gamma}$ results in a tree $\mathcal{T}_+$ that unfolds the event tree $h(\Upsilon_k)$ from each leaf node $l_i \in \mathcal{T}$ such that $l_i \in \Upsilon_k$. If $\mathcal{T} = \emptyset$ and $h : \emptyset \to \Gamma = \{\mathcal{T}_*\}$, then $\boldsymbol{\mathcal{T} \uplus_h \Gamma = \mathcal{T}_*}$.

Now for every time-slice t take a partition $\Upsilon(\mathcal{T}_t)$ associated with a finite event tree $\mathcal{T}_t$ together with a set of finite event trees $S(\mathcal{T}_t)=\{\mathcal{T}_{t_i}; i=1,\ldots,n_t\}$. Next define any map $h_t : \Upsilon(\mathcal{T}_t) \to S(\mathcal{T}_t)$. Note that it is possible that $\mathcal{T}_{t_i} = \emptyset$. Then, for every time-slice $t = 0, 1, \ldots$, we have that

$$\mathcal{T}_{t+1} = \mathcal{T}_t \uplus_{h_t} S(\mathcal{T}_t), \tag{6.1}$$

where $\mathcal{T}_{-1} = \emptyset$ and $h_{-1} : \emptyset \to \{\mathcal{T}_0\}$ whenever there is no time-invariant covariate. Observe that this merging operation corresponds to adding the process-driven objects $\mathcal{T}_1(s_i)$ to the leaf vertex s_i of $\mathcal{T}_0$ in the way depicted in Figure 6.4.

Thus and more formally I define an infinite tree $\mathcal{T}_\infty$ as the direct limit

$$\mathcal{T}_\infty = \varinjlim \mathcal{T}_t \tag{6.2}$$

of the system $\{\Gamma, f(i,j); i,j = -1, 0, 1, \ldots\}$, where $\Gamma = \{\mathcal{T}_t; t = -1, 0, 1, \ldots\}$ and the morphism $f : \mathcal{T}_i \to \mathcal{T}_j, j \geq i$ is such that

$$f(\mathcal{T}_j) = \mathcal{T}_i \uplus_{h_i} S(\mathcal{T}_i) \ldots \uplus_{h_{j-1}} S(\mathcal{T}_{j-1}). \tag{6.3}$$

An important type of infinite event trees called the Periodic Event Tree is defined below. Next I will introduce some useful subclasses of Periodic Event Trees. These families of event trees support processes commonly found in real-world applications because they enable us to embed Markov assumptions and time-homogeneity hypotheses. In Chapter 7 I will use them to define a new class of DCEGs. For a more extensive discussion of periodicity in probabilistic trees, see Peres (1999).

Let $\Lambda(\mathcal{T}) = \{\lambda \subset \mathcal{T}\}$ denote the set of paths of an infinite tree $\mathcal{T}$ where the path λ is either a root-to-leaf or an infinite path of $\mathcal{T}$. Next let $s(t)$ denote a situation that happens in time-slice t. Finally, let $\boldsymbol{\tau}(\lambda) = (\tau_i(\lambda))_{i\in\mathcal{I}(\lambda)}$ denote the ordered sequence of time-slices $\tau_i(\lambda), i \in \mathcal{I}(\lambda)$, associated with each event in a path λ, where $\mathcal{I}(\lambda)$ is the set of indexes corresponding to the vector $\boldsymbol{\tau}(\lambda)$. For instance, define λ as the s_0-to-s_{300} path in Figure 6.4. It then follows that $\mathcal{I}(\lambda) = \{1, \ldots, 6\}$ and $\boldsymbol{\tau} = (0, 0, 0, 1, 1, 1)$.

Definition 31 (T-Periodic Event Tree)**.** An infinite event tree is a T-*Periodic Event Tree*, $T = 0, 1, \ldots$, if and only if for every situation $s_a(t_a), t_a \geq T+1$, there is a situation $s_b(t_b), t_b \leq T$, such that there exists a bijection

$$\phi(s_a, s_b) : \Lambda(\mathcal{T}(s_a)) \rightarrow \Lambda(\mathcal{T}(s_b)), \tag{6.4}$$

satisfying the following two conditions:

1. the ordered sequence of events in a path $\lambda \in \Lambda(\mathcal{T}(s_a))$ equals the ordered sequence of events in the path $\lambda' = \phi(s_a, s_b)(\lambda) \in \Lambda(\mathcal{T}(s_b))$.

2. for every path $\lambda \in \Lambda(\mathcal{T}(s_a))$ we have that $\tau_i(\lambda) = \tau_i(\lambda') + (t_a - t_b)$, $i \in \mathcal{I}(\lambda)$, where $\lambda' = \phi(s_a, s_b)(\lambda) \in \Lambda(\mathcal{T}(s_b))$.

A T-Periodic Event Tree is said to be a *Strong T-Periodic Event Tree* if either $\mathcal{T}_{-1} = \emptyset$ or whenever every subtree $\mathcal{T}_\infty(s), s \in l(\mathcal{T}_{-1})$, that unfolds from each leaf node of $\mathcal{T}_{-1}$ is a T-Periodic Event Tree.

Take a finite event tree $\mathcal{T}$ and define a partition $\Upsilon_a(\mathcal{T}) = \{\Upsilon_{a1}\}$ and a map $h_a : \Upsilon_a(\mathcal{T}) \rightarrow \{\mathcal{T}\}$ – or $\Upsilon_b(\mathcal{T}) = \{\Upsilon_{b1}, \Upsilon_{b2}\}$ and $h_b : \Upsilon_b(\mathcal{T}) \rightarrow \{\mathcal{T}, \emptyset\}$. When each non-empty component of the forest $\mathcal{F}o_T \subset \mathcal{T}_\infty$ is equal to $\mathcal{T}$ the infinite event tree $\mathcal{T}_\infty$ will be called a:

1. T-Periodic Event Tree ($\mathcal{T}$), if $S(\mathcal{T}_t) = \{\mathcal{T}\}$ and $h_t = h_a$ for all $t = T+1$, $T+2, \ldots$; or
2. T-Terminated Periodic Event Tree, if $S(\mathcal{T}_t) = \{\mathcal{T}, \emptyset\}$ and $h_t = h_b$ for all $t = T+1, T+2, \ldots$. In this case, $\Upsilon_{b1}(\mathcal{T}_t)$ is the set of leaf nodes of $\mathcal{T}_t$ whose associated sequence of events at time t corresponds to some leaf node of $\mathcal{T}$ in $\Upsilon_{b1}(\mathcal{T})$, and $\Upsilon_{b2}(\mathcal{T}_t) = l(\mathcal{T}_t) - \Upsilon_{b1}(\mathcal{T}_t)$.

In a T-Periodic Event Tree ($\mathcal{T}$) any branch that unfolds from a situation $s(T)$ at the beginning of time T has infinite length. This implies that a unit at situation $s(T)$ will never arrive at a leaf node. Conversely a process represented by a T-Terminated Periodic Event Tree ($\mathcal{T}$) has some of its branches missing after time T and so a unit at the beginning of time T can take a path that leads it to a leaf node. Note that every T-Periodic Event Tree ($\mathcal{T}$) and T-Terminated Periodic Event Tree ($\mathcal{T}$)

are also Strong T-Periodic Event Trees. For example, the Event Tree in Figure 6.4 is a Strong 0-Periodic Event Tree and a 0-Terminated Periodic Event Tree ($\mathcal{T}$), where $\mathcal{T}$ is depicted in Figure 6.2. I next define another useful kind of event tree named the Laminated Event Tree. As I will discuss in the next chapter, a laminated event tree has a strong link with the DBNs.

Definition 32 (Laminated Event Tree)**.** An Event Tree is called a *Laminated Event Tree* if it satisfies the following two conditions:

1. For any two situations s_a and s_b at the same level, there is an isomorphism between the labels associated with the florets $\mathcal{F}(s_a)$ and $\mathcal{F}(s_b)$.
2. If two situations are at the same level, then they are also in the same time-slice.

6.2 The Probability Space and the Staged Tree

To obtain a staged tree it is necessary to associate a probability space with a given event tree. To do this, take a sample space defined as the set of paths $\Lambda(\mathcal{T}_\infty)$. As for a finite tree (Shafer, 1996), a situation s_i in an infinite tree can be associated with a random variable $X(s_i)$ that describes the possible developments of a process once a unit has arrived at s_i. The state space $\mathbb{X}(s_i)$ of $X(s_i)$ corresponds to the set of emanating edges $(s_i, s_j), s_j \in ch(s_i)$, of s_i. Now define, for each situation $s_i \in \mathcal{T}_\infty$, the primitive probabilities

$$\pi(s|s_i) = P(X(s_i) = s|s_i), s \in ch(s_i). \tag{6.5}$$

Let the path-cylinder $\Lambda(s_i)$ be the set of all paths in $\Lambda(\mathcal{T}_\infty)$ that pass through a situation s_i. Let $\Psi(s_i)$ be the time-ordered concatenation of situations along the root-to-$pa(s_i)$ path. Also let $\psi(s, s_i), s \in \Psi(s_i)$, denote the child situation of s along the root-to-s_i path. From the usual rules of conditioning, we then have that

$$P(\Lambda(s_i)) = \prod_{s \in \Psi(s_i)} \pi(\psi(s, s_i)|s). \tag{6.6}$$

Using the Extension Theorem (Feller (1971b), p. 119), we are then able to uniquely extend the probability measure defined in Equation 6.6 to the smallest σ-algebra of $\{\Lambda(s_i); s_i \in \mathcal{T}_\infty\}$, the so called path-cylinder σ-algebra; see also Segala (1995) and Stoelinga (2002). So our probabilistic model is well specified by a pair $(\mathcal{T}_\infty, \Pi)$, where

$$\Pi = \{\pi(s|s_i); s \in ch(s_i), s_i \in \mathcal{T}_\infty)\} \tag{6.7}$$

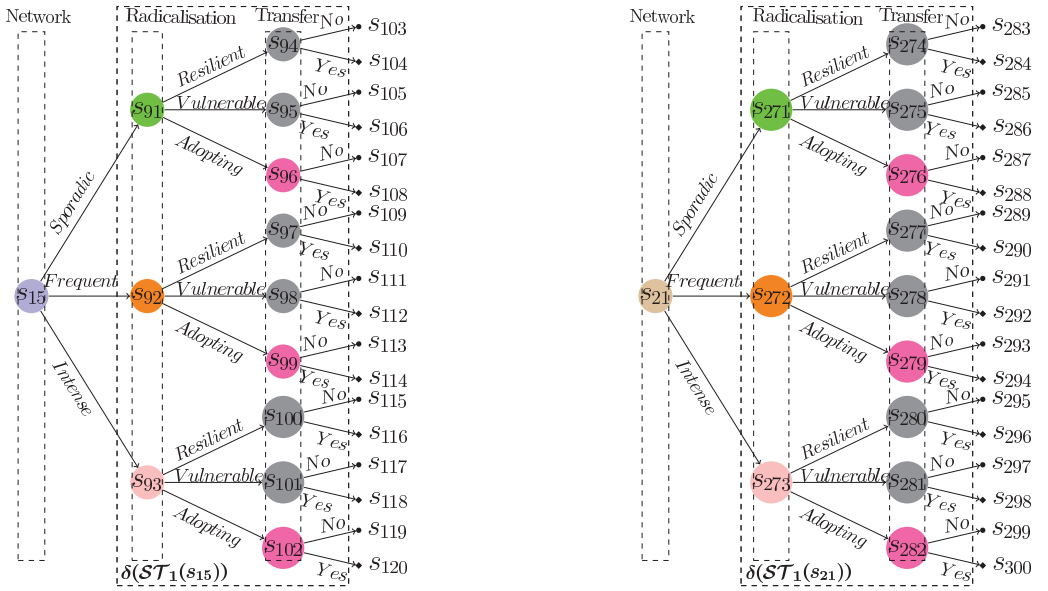
is the set of all primitive probabilities defined over an infinite event tree $\mathcal{T}_\infty$.

Completely analogous to a finite event tree (Smith and Anderson, 2008), we say that in an infinite event tree two situations s_a and s_b are in the same stage u if and only if the probability distributions of their corresponding random variables $X(s_a)$ and $X(s_b)$ are identical under a bijection

$$\phi_u(s_a, s_b) : \mathbb{X}(s_a) \to \mathbb{X}(s_b). \tag{6.8}$$

So a stage is a partition set of situations that collects together in a single cluster all those situations whose 1-step unfoldings are exchangeable conditional on the event that a unit is at a situation contained in this stage. As in a finite staged tree, every stage in an infinite staged tree has a unique colour that differentiates it from others. A staged tree $\mathcal{ST}_\infty$ is then obtained when its corresponding event tree $\mathcal{T}_\infty$ is embellished with those colours.

Example 3 (Dynamic Radicalisation Process - cont.). Return to the radicalisation process described in Example 3 and depicted in Figure 6.4. Figures 6.6a and 6.6b depict the staged subtrees for the first time-slice with respect to vulnerable non-transferred prisoners who keep, respectively, sporadic (situation s_{15}) and frequent (situation s_{21}) social contact with extremist recruiters during the initial time-slice. Here, for example, since the probability of transfer is conditionally independent of social interactions and does only change for prisoners currently adopting radicalisation, the situations associated with the variable Transfer can be coloured using only two colours, grey and pink. It then follows that this model has only two different stages associated with the set of situations corresponding to the variable Transfer.



(a) Staged Subtree $\mathcal{ST}_1(s_{15})$

(b) Staged Subtree $\mathcal{ST}_1(s_{21})$

Figure 6.6: Two Staged Subtrees of time-slice 1 corresponding to the dynamic radicalisation process described in Example 3 and depicted in Figure 6.4.

In contrast to a DBN we are able now to represent directly in the staged tree the context-specific statement corresponding to the transfer dynamic. Observe that by retaining the colour consistency between the process-driven objects we are able to analyse different branches of the process and compare them without having to draw the whole Staged Tree $\mathcal{ST}_\infty$. Also note that from these coloured trees we can see that the only difference between the two processes in time-slice 1 with respect to prisoners at situations s_{15} and s_{21} is in terms of the membership of their networks. This is because their corresponding situations are the only ones that have different colours (blue and brown). □

I have noted that so called decision trees (Friedman and Goldszmidt, 1998) were used to explore context-specific local structures in BNs. Using a similar recursive propositional structure presented in Equation 6.1, a decision tree is constructed for each variable Z with which there are context-specific conditional independences associated in a BN. A decision tree is elicited only using the set of random variables that are parents of Z. If a BN has context-specific independences corresponding

to two or more variables, then it will be necessary to draw a set of decision trees to depict them. In contrast to this a stage tree depicts all context-specific conditional independences using a single tree graph that represents the whole process. Also remember that staged trees do not demand to define a process using a pre-defined set of random variables. In this sense, a staged tree captures broader local structures and provides us with a more flexible frame to concatenate the context-specific statements.

A staged tree whose states at time-slice $t, t \geq T$, only depend on events that happens in the last N previous time-slices and the current one is called a N-Markov Staged Tree after time T. If time-homogeneous conditions are enforced over the primitive probabilities, we then obtain a time-homogeneous Staged Tree. The staged tree associated with Example 3 is 1-Markov since the current state of the prison system is assumed to be affected only by the events that happened 1-step ago and by the actions taken in it. Also note that this staged tree is time-homogeneous in a sense formally defined below. Thus, let $\xi(s, k)$ be the time-ordered concatenation of events that precedes a situation s in the last k time-slices and in the time-invariant tree $\mathcal{T}_{-1}$. So, for example, in Figure 6.6a (see also Figure 6.4) since there is no time-invariant event tree $\mathcal{T}_{-1}$ we have that

- $\xi(s_{95}, 0) = (Sporadic, Vulnerable)$ and
- $\xi(s_{95}, 1) = (Sporadic, Vulnerable, No, Sporadic, Vulnerable)$.

Definition 33 (N-Markov Staged Tree after time T)**.** A staged tree is called an *N-Markov Staged Tree after time T* if there is a time-slice $T, T \geq N$, such that for every time-slice $t, t \geq T$, we have that

$$\pi(s|s_i) = P(X(s_i) = s|s_i) = P(X(s_i) = s|\xi(s_i, N)), \tag{6.9}$$

where $s \in ch(s_i)$. If T is equal to N, the staged tree is simply called *N-Markov Staged Tree*. A *Markov Staged Tree after time T* is an N-Markov Staged Tree after time T for some $N \leq T$.

Definition 34 (N Time-Homogeneous Staged Tree after time T)**.** An *N Time-Homogeneous Staged Tree after time T* is an N-Markov Staged Tree after time T

whose corresponding event tree is strong T-periodic and for every situations s_a and s_b, such that s_a and s_b are, respectively, situations in time-slice $t_a \geq T$ and $t_b \geq T$, we have that

$$\xi(s_a, N) = \xi(s_b, N) \Rightarrow \pi(s|s_a) = \pi(s|s_b). \tag{6.10}$$

If T is equal to N, then the staged tree is simply called an N *Time-Homogeneous Staged Tree*. A *Time-Homogeneous Staged Tree after time* T is an N Time-Homogeneous Staged Tree after time T for some $N \leq T$.

A Time-Homogeneous Staged Tree after time T based on a T-Periodic Event Tree ($\mathcal{T}$) or a T-Terminated Periodic Event Tree ($\mathcal{T}$) is an important type of staged trees because it can be constructed using a finite number of coloured process-driven objects. Periodicity yielded by $\mathcal{T}$ and time-homogeneity imply that at the end of every time-slice $t, t \geq T$, there is only one finite set of colourful subtrees $S(\mathcal{ST}_t) = S$ that can unfold in the next time-slice $t+1$. The map $h_t : \Upsilon(\mathcal{ST}_t) \rightarrow S(\mathcal{ST}_t)$ that specifies which finite staged subtree unfolds from each path of $\mathcal{ST}_t$ has also a repeating structure determined by these two conditions. Formally, for all $t, t \geq T$, there is a bijection $\phi_t : \Upsilon(\mathcal{ST}_t) \rightarrow \Upsilon(\mathcal{ST}_T)$ such as $h_t = h_T \circ \phi_t$.

I next define another important class of staged trees called Laminated Staged Tree (Definition 35) that is supported by laminated event trees (Definition 32). For this purpose, recall from Section 2.2 that two situations in an event tree whose distances from the root node are the same and equal to d are said to be at the same *level* ℓ_d. Now take a partition $\mathscr{L} = \{\mathscr{L}_i\}$ of its levels such that any two levels in the same set $\mathscr{L}_i$ are at different time-slices. The partition $\mathscr{L}$ is then said to be a *laminated level partition*. This often happens when a stage structure has a strong link with the variables defining the problem.

For instance, in Example 3 there is an implicit assumption that we can only use the same colour for situations associated with the same variable. We then have that $\mathscr{L} = \{\mathscr{L}_0, \mathscr{L}_1, \mathscr{L}_2\}$, where $\mathscr{L}_i = \{\ell_{i+3j}; j = 0, 1, 2, \ldots\}$. In this case, the sets of levels $\mathscr{L}_0, \mathscr{L}_1$ and $\mathscr{L}_2$ are defined, respectively, by the variables N, R and T. So we use the colour orange (Figures 6.6a and 6.6b) only for situations (s_{92}, s_{272})

corresponding to the variable R. Note that there are situations in every time-slice $t, t \geq 1$, that must also be coloured orange due to the time-homogeneity of the process.

Definition 35 (Laminated Staged Tree)**.** A Staged Tree is said to be a *Laminated Staged Tree* if and only if the following two conditions hold:

1. Its corresponding event tree is laminated.
2. There is a a laminated level partition $\mathscr{L}$ such that each stage only merges situations associated with levels in the same partition set $\mathscr{L}_i \in \mathscr{L}$.

Finally, two situations s_a and s_b are said to be in the same *position* w in an infinite staged tree if and only if the bijection in Equation 6.4

$$\phi(s_a, s_b) : \Lambda(\mathcal{T}(s_a)) \rightarrow \Lambda(\mathcal{T}(s_b))$$

satisfies an additional condition that the ordered sequence of colours in a path λ, $\lambda \in \Lambda(\mathcal{T}(s_a))$, is identical to the ordered sequence of colours in the path λ', such that

$$\lambda' = \phi(s_a, s_b)(\lambda) \in \Lambda(\mathcal{T}(s_b)).$$

Below I define a particular type of periodicity in infinite staged trees using the concept of position. It is straightforward to verify that every T-Periodic Staged Tree is a T-Periodic Event Tree.

Definition 36. An infinite staged tree is a T-*Periodic Staged Tree*, $T = 0, 1, \ldots$, if and only if for every situation $s_a(t_a), t_a \geq T+1$, there is a situation $s_b(t_b), t_b \leq T$, such that both situations are in the same position.

6.3 Obtaining a Dynamic Chain Event Graph

Analogous to a CEG, a DCEG is constructed by merging all situations in the same position into a single vertex and then gathering all situations that represent terminated processes into a single position w_∞. Every staged tree then spans a unique DCEG by vertex contraction operations over the set of situations.

Note that the definition of position is more restrictive in a DCEG than in a CEG because of the condition 2 associated with the bijection in Equation 6.4 (see Definition 31). So, whilst in a CEG any two situations are at the same position when their unfolding sequences of events and colour are equivalent, in a DCEG these equivalences of events and colours have additionally to hold in each time-slice. This is important to avoid topological ambiguities during the final construction of a DCEG graph since the set of vertices in a DCEG graph corresponds exactly to the set of positions yielded by its supporting stage tree.

Also observe that two processes unfolding from situations in the same position must be equivalent for all subsequent developments described by the staged tree. However processes evolving from situations in the same stage only have to be identified across the next step in their evolutions. Therefore, as in a CEG the set of positions in a DCEG constitutes a finer partition of its corresponding staged structure: if two situations are in the same position, then they must also be in the same stage but the converse is not always valid.

As observed in Barclay et al. (2015) a DCEG may have directed loops that allow it to have a finite number of vertices. Theorem 7 below tells us that this is always the case when a DCEG is based on a T-Periodic Staged Tree. Theorem 8 asserts that time-homogeneity suffices to satisfy this condition.

Theorem 7. *A DCEG is finite if and only if its corresponding staged tree is T-Periodic after some time T.*

Proof. Assume that a finite DCEG is supported by a staged tree that it is non-periodic for every $T, T = 0, 1, \ldots$. It will then follow that for every time-slice T, $T = 1, 2, \ldots$, there must be at least one situation $s_a(T)$ that is at position w, such that w does not merge any situation $s_b(t)$, $t = 0, \ldots, T-1$. Therefore the number of positions in this staged tree is infinite. This contradicts the hypothesis. It follows that the supporting staged tree of a finite DCEG must therefore be T-Periodic Staged Tree for some T.

Conversely if a stage tree is T-periodic, then for every situation $s(t_a)$ at time t_a,

$t_a = T+1, T+2, \ldots$, there is some situation $s(t_b)$ at time t_b, $t_b, t = 0, \ldots, T$, such that s_a and s_b are at the same position. So the number of positions in the corresponding DCEG does not need to be greater than the number of situations in $\mathcal{T}_T$. This DCEG must therefore be finite. ■

Theorem 8. *Every time-homogeneous staged tree after time T has an associated DCEG with a finite graph.*

Proof. In any event tree, let $\xi(s_a, s_b)$ be the sequence of events that happen along a finite path between the situations s_a and s_b, where s_b is down stream of s_a. Let $\mathcal{S}(t+1)$ be the set of all situations in times-slice $t+1$ whose parent are in time-slice t. Also let $a(s_i, t)$ be the antecedent situation of a situation s_i such that $a(s_i, t) \in \mathcal{S}(t)$.

By assumption, the event tree is strong T-periodic. Initially assume that $\mathcal{T}_{-1} = \emptyset$. So for every situation $s_i \in \mathcal{S}(2T+2)$ there is a situation $s(t), t \leq T$, such that the event trees $\mathcal{T}(a(s_i, T+1))$ and $\mathcal{T}(s(t))$ are graphically isomorphic. It follows that there is a situation $s(t^*) \in \mathcal{T}(s(t))$, $t^* = t+T+1$, such that $s(t^*) \in \mathcal{S}(t^*)$, $\xi(s(t), s(t^*)) = \xi(a(s_i, T+1), s_i)$ and the event trees $\mathcal{T}(s_i)$ and $\mathcal{T}(s(t^*))$ are graphical isomorphic. Time-homogeneity after time T then tell us that there is also a probabilistic isomorphism between the primitive probabilities associated with $\mathcal{T}(s_i)$ and $\mathcal{T}(s(t^*))$. Therefore, s_i and $s(t^*)$, where $t^* \leq 2T+1$, are in the same position. So the number of positions in the corresponding DCEG does not need to be greater than the number of situations in $\mathcal{T}_{2T+1}$. This DCEG must therefore be finite.

If there are time-invariant events ($\mathcal{T}_{-1} \neq \emptyset$), the result still holds for each subtree $\mathcal{T}_\infty(s), s \in l(\mathcal{T}_{-1})$, because of the strong periodicity embedded into the time-homogeneity condition. Therefore the DCEG is also finite in this case. ■

Figure 6.7 shows the finite DCEG associated with Example 3. Note that to draw an uncluttered graph without any loss, some of its edges are dashed and grey and the sink position w_∞ is represented by *two* receiving vertices. In the next chapter

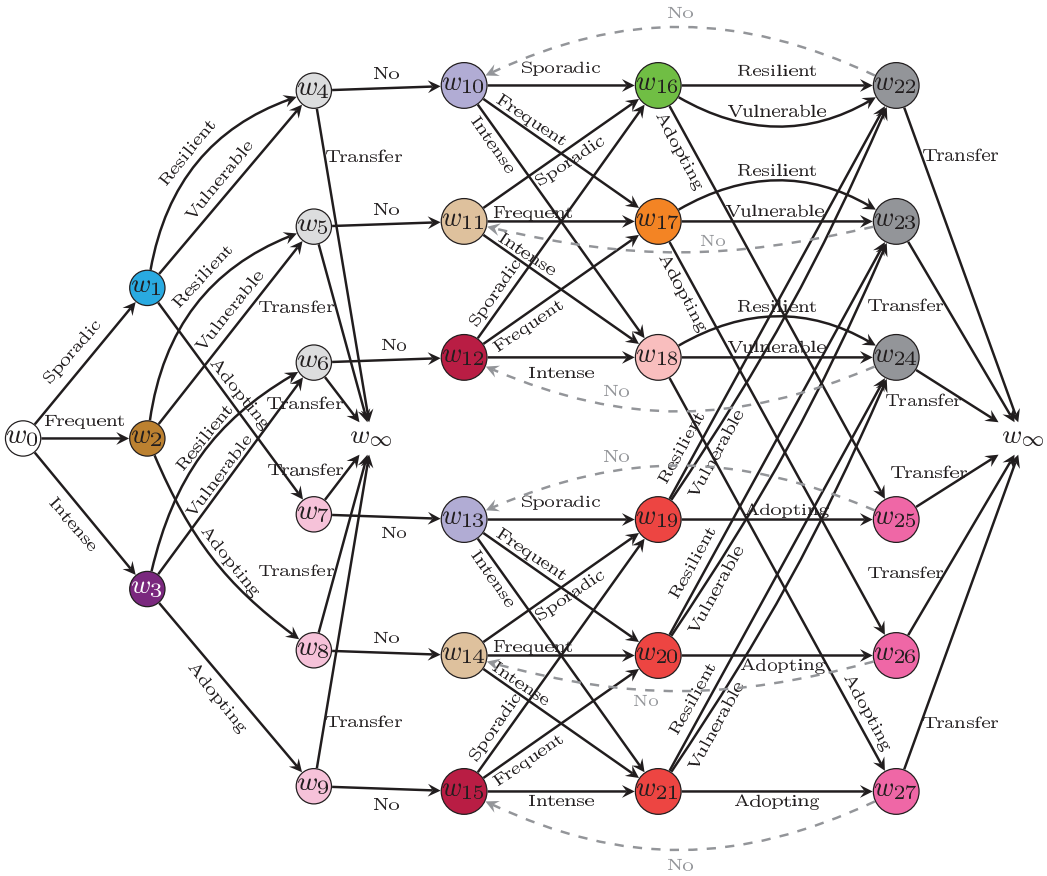


Figure 6.7: The 2T-DCEG associated with Example 3. The stage structure is given by the following partition: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3\}$, $u_4 = \{w_4, w_5, w_6\}$, $u_5 = \{w_7, w_8, w_9\}$, $u_6 = \{w_{10}, w_{13}\}$, $u_7 = \{w_{11}, w_{14}\}$, $u_8 = \{w_{12}, w_{15}\}$, $u_9 = \{w_{16}\}$, $u_{10} = \{w_{17}\}$, $u_{11} = \{w_{18}\}$, $u_{12} = \{w_{19}, w_{20}, w_{21}\}$, $u_{13} = \{w_{22}, w_{23}, w_{24}\}$, $u_{14} = \{w_{25}, w_{26}, w_{27}\}$.

I will define a useful family of finite DCEGs named N Time-Slice Dynamic Chain Event Graph (NT-DCEG). It will become apparent later that the DCEG given in Figure 6.7 is indeed a 2T-DCEG. At that point I will explain how to read the conditional independences embedded in its topology.

Chapter 7

An N Time-Slice Dynamic Chain Event Graph

In this chapter I will define a *novel subclass* of DCEGs called the N Time-Slice DCEG (NT-DCEG) and proceed to investigate some its properties. A close link between an NT-DCEG and a Markov process will be first provided. I will then explore some connections between DCEGs and DBNs (Section 2.4). In particular, I will prove that the 2T-DBNs constitute a special family of 2T-DCEGs.

I will next develop a filtration of the NT-DCEG σ-algebra using an appropriate set of CEGs. This will equip us with the theoretical background to show that an NT-DCEG enables us to consider highly asymmetric processes embodying certain classes of Granger causal hypotheses. I will end this chapter by demonstrating how to explore the topology of an NT-DCEG graph in order to define useful random variables and obtain some separation theorems that apply to these variables.

7.1 The Semantics of the NT-DCEG

Definition 37 below introduces the concept of a T-position. This demands a further constraint on the definition of a regular position. So situations in a particular T-position must also be in the same position but the converse is not always valid.

Definition 37. Two situations $s_a(t_a)$ and $s_b(t_b)$ are in the same $\boldsymbol{T}$**-position** if

and only if they are in the same position, and $t_a, t_b \geq T$ or $t_a = t_b < T$.

A T-position avoids cycles before a time-slice T whilst preserving all other characteristics of a standard DCEG. Using this construction we can demand that a finite DCEG has all its loops rooted at situations that happen at the same time-slice if its staged tree is time-homogeneous after some time T.

Based on the concept of T-positions, we can now define a useful DCEG class, called the N Time-Slice Dynamic Chain Event Graph (NT-DCEG). This has a unique periodic graphical structure over all time-slices and its primitive probabilities are all time-homogeneous for time-slices $t, t \geq N$.

Theorem 8 guarantees that an NT-DCEG is also a finite graph. Note that in many real-world applications a time-slice T might exist with the property that it is possible to obtain the same graphical model regardless of whether the nodes of the graph represented a position or a T-position. In Example 3 this in fact is the case if we adopt $T=0$ or $T=1$. The standard DCEG and a 2T-DCEG (Figure 6.7) that each represent the radicalisation process will then be identical.

Definition 38. A DCEG defined in terms of $(N-1)$-positions is called an ***N* Time-Slice Dynamic Chain Event Graph** (NT-DCEG) if and only if the following conditions hold:

1. Its event tree is 0-Periodic Event Tree $(\mathcal{T})$ or 0-Terminated Periodic Event Tree $(\mathcal{T})$; and
2. Its staged tree is time-homogeneous after time $(N-1)$.

Recall that an event tree $\mathcal{T}_{-1}$ is associated with time-invariant covariates and an event tree $\mathcal{T}$ fully characterises every time-slice of a model obtained from 0-Periodic Event Tree $(\mathcal{T})$ or 0-Terminated Periodic Event Tree $(\mathcal{T})$ (see Section 6.1). Therefore, an NT-DCEG requires us to elicit only two finite process-driven objects: $\mathcal{T}_{-1}$ and $\mathcal{T}$. To obtain a staged tree, because of the time-homogeneous condition it is necessary to define explicitly only those primitive probabilities associated with the first N time-slices.

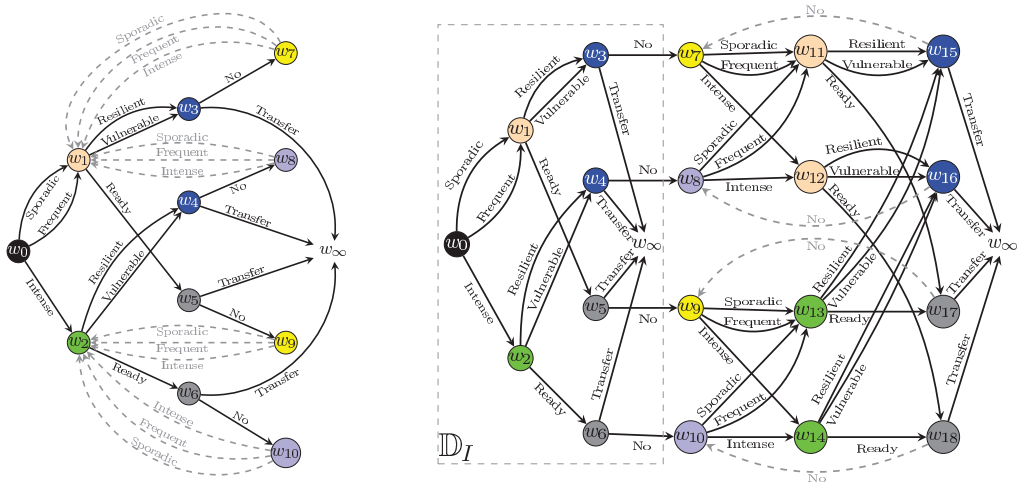
From a graphical point of view the use of a $N-1$-position concept enables us to enforce loops only from time-slice $N-1$ on. This is important when we need to merge DCEGs that are spanned by different branches of the same event tree. For example, based on an event-tree that splits the process according to some time-invariant covariates a distributed model construction can compose the domain information coherently. In this case, every sub-process has its own particular DCEG model. To merge these DCEGs and so to stress common periodic characteristics that may be shared between them, it is helpful to demand that all loops must be rooted at situations that happen at the same time-slice. Although this condition is not strictly necessary it does facilitate the readability of the final DCEG and the design of efficient algorithmic structures. These points are further discussed in the example below.

Example 9 (Extended Dynamic Radicalisation Process - cont.). Assume that in Example 9 the whole prison dynamic is independent of prisoners' nationality. This process can be expected to be driven by cultural and social factors. The nationality is then not an appropriate explanatory variable. Assume that the radicalisation process of a prior convicted prisoner is represented by the 2T-DCEG depicted in Figure 6.7 that embedded all the hypotheses described in Example 9.

The radicalisation dynamic of a prisoner having no criminal antecedents is now shown by two different models in Figure 7.1. These models, a DCEG (Figure 7.1a) and a 2T-DCEG (Figure 7.1b), are equivalent. This is because they depict the same set of conditional independence statements. In line with domain experts' information, this structure is simpler than that of a prisoner with previous convictions in a sense that additional context-specific conditional independences to those hypothesised in Example 3 apply to this case.

Also the radicalisation risk of a non-prior convicted prisoner is lower than a prior convicted prisoner with similar behaviour pattern in the prison. Here the colours used in both 2T-DCEGs depicted in Figures 6.7 and 7.1b are compatible. In other words, if a position w_a in Figure 6.7 and a position w_b in Figure 7.1b have the same colour, they are then in the same position in a merged 2T-DCEG model.

Figure 7.1a shows a DCEG where some situations in the initial time-slice are in positions that also aggregate situations that unfold in the subsequent time-slices. This enables us to obtain a very simple graph with only four levels.



(a) A DCEG

(b) A 2T-DCEG

Figure 7.1: A DCEG and a 2T-DCEG corresponding to the radicalison process of prisoners without prior criminal convictions in Example 9.

Nevertheless this simplification has some drawbacks in terms of the readability of the conditional independences represented by the model. For example, position w_2 *corresponds to a inmate who has intense social contacts with other radicalised prisoners during the initial time-slice. It also merges situations in time-slice* t*,* $t = 1, 2, \ldots$*, that correspond to a radicalised inmate in the previous time-slice* $t - 1$ *who remains in prison at the current time-slice* t*. This last statement cannot be read immediately from the DCEG presented in Figure 7.1a. In contrast, we can read it directly from its corresponding 2T-DCEG presented in Figure 7.1b. Adopting the concept of a* 1*-position thus has enabled us to conveniently construct a subgraph* $\mathbb{D}_I$ *associated with the initial time-slice that can act as a legend to analyse the subsequent time-slices.*

Observe also that it is easier to compare the radicalisation processes of a prior and a non-prior convicted inmates using the 2T-DCEG than using the DCEG depicted in Figure 7.1. It is simple to verify that in this case we obtain the 2T-DCEG illustrated in Figure 7.2. Note that the use of the DCEG in Figure 7.1a to merge

both radicalisation process would not be so direct.

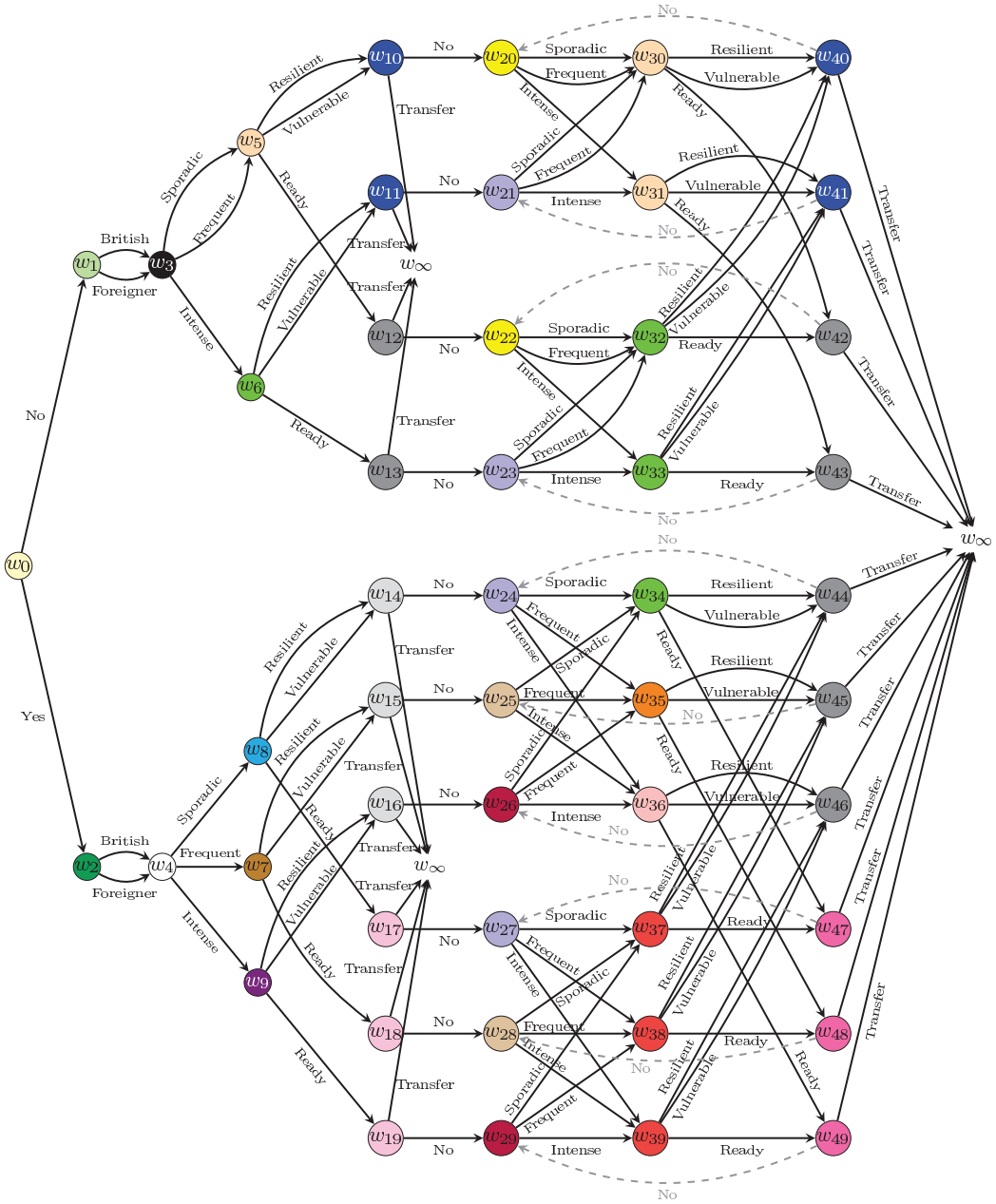


Figure 7.2: The 2T-DCEG associated with Example 9. Hotter colours implies higher risk of radicalisation. □

It is useful at this stage to introduce the definition of temporal edge in a DCEG $\mathbb{C}$. Note that a temporal edge associated with time $N-1$ will also be a temporal edge associated with every time $t, t = N, N+1, \ldots$. This happens because of the time-homogeneity condition required from every NT-DCEG. These temporal edges – called cyclical temporal edges below – enable us to represent a time-

homogeneous map that connects positions in two consecutive time-slices.

Definition 39 (Temporal Edge)**.** Take a DCEG $\mathbb{C}$ obtained from an infinite tree $\mathcal{T}_\infty$. A directed edge (w_a, w_b) in $\mathbb{C}$ is a *temporal edge* associated with time-slice t if and only if for some time t there exist two situations $s_a \in w_a$ and $s_b \in w_b$ such that $s_b \in ch(s_a)$, $ch(s_b) \neq \emptyset$ and $s_b \in l(\mathcal{T}_t)$, where $\mathcal{T}_t \subset \mathcal{T}_\infty$. We call a temporal edge associated with time-slices t, $t = N-1, N, \ldots$, of $\mathbb{C}$ a *cyclical temporal edge*. Henceforth we will denote the set of cyclical temporal edges by $E_{\textcircled{\dagger}}$. We will also define $\mathcal{W}_{Head}$ as the set of positions for which for every position $w \in \mathcal{W}_{Head}$ there exists an edge $(w^*, w) \in E_{\textcircled{\dagger}}, w^* \in \mathbb{C}$.

Theorem 9 tells us that every NT-DCEG can be interpreted as a Markov Chain with state space $\mathcal{X} = \mathcal{W}_{Head}$, if the underlying event tree is a 0-Periodic Event Tree $(\mathcal{T})$, or alternatively if $\mathcal{X} = \mathcal{W}_{Head} \cup \{w_\infty\}$, in which case the inherent event tree is a 0-Terminated Periodic Event Tree $(\mathcal{T})$. This is an important link enabling the corresponding DCEG to be represented in a very compact way.

Focusing only on the transitions between time-slices, the Markov Chain projection now provides a framework for domain experts to analyse how the system may develop over time. For example, domain experts can explore the equilibrium state of the Markov Chain and can also obtain the respective rate of convergence to it given the actual state of the process. These analytical results may suggest the necessity of some systemic intervention. In order to perform such an exploration it can be helpful to zoom in again over the conditional independences depicted into the corresponding DCEG. Corollary 8 guarantees that a Markov Chain spanned by an NT-DCEG based on a Periodic Event Tree $(\mathcal{T})$ without time-invariant events and all of whose primitive probabilities are strictly positive has a stationary distribution. In contrast, an NT-DCEG yielded by Terminated Periodic Event Tree $(\mathcal{T})$ always has at least one absorbing state because $w_\infty \in \mathcal{X}$.

To construct such a Markov Chain, let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{|\mathcal{X}|})$ be its initial distribution, where $\mu_i, i = 1, \ldots, |\mathcal{X}|$, is the probability of a position $w_i \in \mathcal{W}_{Head}$ to be reached at the end of $N-1$ time-slices. By convention μ_1 is always associated with the

position w_∞ if $w_\infty \in \mathcal{X}$. In other words, each u_i is equal to the sum of the occurrence probabilities associated with each w_0-to-w_i path in a DCEG. This can then be translated into the sum of occurrence probabilities associated with each root-to-$s_j(N-1), s_j \in w_i$, path in the event tree. We therefore have that

$$\mu_i = \sum_{s_a(N-1)\in w_i} P(\lambda(s_0, s_a)) = \sum_{s_a(N-1)\in w_i} \prod_{s\in\Psi(s_a)} \pi(\psi(s, s_a)|s), \qquad (7.1)$$

where $\lambda(s_a, s_b)$ denotes the s_a-to-s_b path in the event tree.

Now define $\boldsymbol{M} = [m_{ij}]$ as a transition matrix, where m_{ij} represents the transition probability from a state $x_i \equiv w_i \in \mathcal{X}$ to a state $x_j \equiv w_j \in \mathcal{X}$. Each m_{ij} corresponds to the sum of the probabilities associated with each walk that goes from a position w_i to a position w_j in only one time-slice. If w_j cannot be reached from w_i in one time-slice, or $i = 1$ and $w_\infty \in \mathcal{X}$, then $m_{ij} = 0$. Again, every non-null m_{ij} can be expressed as the following function of primitive probabilities:

$$m_{ij} = \sum_{s_a(N-1)\in w_i} \sum_{s_b(N)\in w_j} P(\lambda(s_a, s_b)) = \sum_{s_a(N-1)\in w_i} \sum_{s_b(N)\in w_j} \prod_{s\in\Psi(s_a,s_b)} \pi(\psi(s, s_b)|s). \qquad (7.2)$$

Theorem 9. *There is a map from every NT-DCEG into a finite state-transition diagram.*

Proof. Construct a Markov Chain whose state space is $\mathcal{X} = \mathcal{W}_{Head}$, if the underlying event tree is a 0-Periodic Event Tree ($\mathcal{T}$), or $\mathcal{X} = \mathcal{W}_{Head} \cup \{w_\infty\}$, if the inherent event tree is a 0-Terminated Periodic Event Tree ($\mathcal{T}$). Take the initial distribution and the transition matrix as given by Equations 7.1 and 7.2, respectively. The state-transition diagram of this Markov Chain is then finite. So the result follows. ■

Corollary 8. *Every NT-DCEG obtained from a 0-Periodic Event Tree ($\mathcal{T}$) whose probability associated with each edge is non-null and $\mathcal{T}_{-1} = \emptyset$ has a corresponding Markov process that is ergodic and irreducible.*

Proof. The strong 1-periodicity of the underlying event guarantees that there is a walk that goes from every position $w_i \in \mathcal{W}_{Head}$ to any position in $\mathcal{W}_{Head}$ in

only one time-slice. Since the primitive probabilities are all positive, every unit in a position $w_i \in \mathcal{W}_{Head}$ has a non-zero probability of returning to the same position w_i or to reach a position $\mathcal{W}_{Head}\backslash\{w_i\}$ in the end of a time-slice. From the definition of the Markov Chain described in the proof of Theorem 9, it can then be seen that the corresponding Markov Chain is ergodic and irreducible. ■

Example 3 (Dynamic Radicalisation Process - cont.). Figure 7.3 depicts the state-transition diagram of the Markov Chain corresponding to the NT-DCEG showed in Figure 6.7. The Markov Chain enables us to present the radicalisation process compactly using just a few positions of the elicited NT-DCEG. Being based on a tree, an NT-DCEG provides domain experts with an intuitive framework not only to represent and estimate a process but also to interpret it using a single random variable whose states over time are given by a finite set of positions. Here the radicalisation process can be explained using a random variable that has seven states represented by the positions $w_{10}, \ldots, w_{15}$ and w_∞.

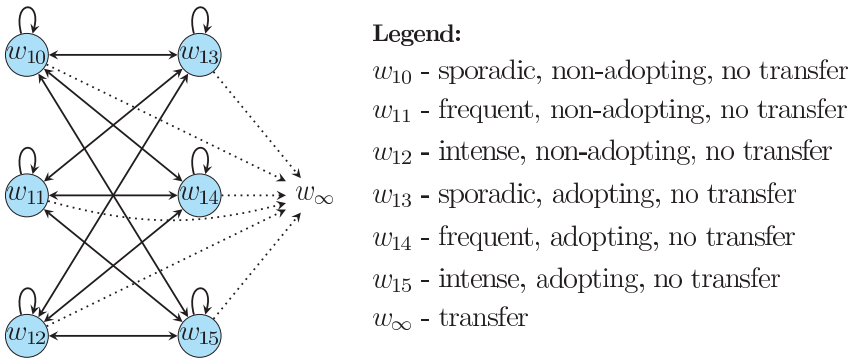


Figure 7.3: The state-transition diagram associated with the 2T-DCEG depicted in Figure 6.7

Note that the state-transition diagram tells us that the categories Resilient and Vulnerable associated with the variable Radicalisation can be merged without losing any useful information for the macro-level interpretation of the chosen NT-DCEG. It also follows directly that all states connected with prison transfers can be represented by only one absorbing state w_∞. In this way, the state-transition diagram provides us with an evocative picture of the overall dynamic development over time.

Furthermore, its transition matrix can be obtained by learning the NT-DCEG. □

7.2 The Relationship between a 2T-DCEG and a 2T-DBN

We next define a Laminated Dynamic Chain Event Graph and prove that every Two Time-Slice DBN (2T-DBN) can be rewritten as a Two Time-Slice Laminated DCEG.

Definition 40 (Laminated Dynamic Chain Event Graph)**.** A DCEG is called a *Laminated Dynamic Chain Event Graph* when it is obtained from a laminated staged tree. If the staged tree of a NT-DCEG is laminated, we obtain an N *Time-Slice Laminated DCEG*.

Let $\boldsymbol{\mathcal{Z}}(-1) = (\mathcal{Z}_1(-1), \ldots, \mathcal{Z}_r(-1))$ be a vector of time-invariant univariate random variables $\mathcal{Z}_i(-1), i = 1, \ldots, r$, and $\boldsymbol{\mathcal{Z}}(t) = (\mathcal{Z}_1(t), \ldots, \mathcal{Z}_s(t))$ be a vector of univariate random variables $\mathcal{Z}_i, i = 1, \ldots, s$, that take values at each time-slice $t = 0, 1, \ldots$. Note that an N Time-Slice Laminated DCEG $\mathbb{C}$ without terminating events has a supporting event tree $\mathcal{T}_\infty$ whose set of paths $\Lambda(\mathcal{T}_\infty)$ can then be expressed as a product space associated with an infinite sequence of random variables $\boldsymbol{\mathcal{Z}}(-1), \boldsymbol{\mathcal{Z}}(0), \boldsymbol{\mathcal{Z}}(1), \ldots$. Therefore the set of paths can be written as

$$\Lambda(\mathcal{T}_\infty) = \boldsymbol{\mathcal{Z}}(-1) \times \boldsymbol{\mathcal{Z}}(0) \times \boldsymbol{\mathcal{Z}}(1) \times \ldots.$$

Obviously, if the N *Time-Slice Laminated DCEG* $\mathbb{C}$ does not have time-invariant random variables, the set of paths is written as

$$\Lambda(\mathcal{T}_\infty) = \boldsymbol{\mathcal{Z}}(0) \times \boldsymbol{\mathcal{Z}}(1) \times \ldots.$$

This framework can be particularly useful for causal analyses when the total or partial variable orders of the vectors $\boldsymbol{\mathcal{Z}}(-1)$ and $\boldsymbol{\mathcal{Z}}(t)$ imply different causal hypotheses. For causal model search using CEGs see Cowell and Smith (2014).

The Laminated DCEG class is also an important model family because Theorem 10 tells us that every DBN can be rewritten as a Laminated DCEG. In particular, according to Theorem 11 every 2T-DBN can be translated into a Two Time-Slice

Laminated DCEG. These results enable us to embellish a DBN with context-specific statements using the broader class of DCEG models. This can be helpful for model search since a DCEG model space is considerably larger than its corresponding DBN model space. For example, the 2T-DBN model selection can be used as a starting point for a 2T-DCEG model search. For an application of such model search strategy using CEGs and BNs, see Barclay et al. (2013). Alternatively, a DCEG model can provide a framework for constructing random variables (see Section 7.6) which in turn enable us to express the context-specific statements using a DBN model.

Theorem 10. *All conditional independence statements entailed by a DBN can be depicted by a Laminated DCEG.*

Proof. Take a DBN $\mathbb{B} = (\bigcup_t V(t), \bigcup_t E(t) \cup E_\dagger(t))$, where $V(t)$, $E(t)$ and $E_\dagger(t)$ are, respectively, the vertex, edge and temporal edge sets associated with time-slice t; see Section 2.4. Denote by $N(t)$ the total number of variables associated with time-slice t. Thus any vertex $v_i(t) \in V(t)$ can be well-defined in the whole vertex set $\bigcup_t V(t)$ by a new index $\jmath = i + \sum_{k=-1}^{t-1} N(k)$, where $N(-1) = 0$.

Let $\mathcal{F}(v_\jmath)$ be a floret associated with the variable represented by the vertex $v_\jmath$ in the DBN $\mathbb{B}$. Define the set of trees

$$\Gamma = \{\mathcal{T}_{(\jmath)}; \mathcal{T}_{(1)} = \mathcal{F}(v_1) \text{ and } \mathcal{T}_{(\jmath)} = \mathcal{T}_{(\jmath-1)} \uplus_{h_{\jmath-1}} \{\mathcal{F}(v_{\jmath-1}), \emptyset\}, \jmath = 2, 3, \ldots\},$$

where $h_\jmath$ is a map from every leaf vertex of $\mathcal{T}_{(\jmath)}$ not associated with a terminating event into the floret $\mathcal{F}(v_\jmath)$ and into $\emptyset$ otherwise. Now take the event tree given by the direct limit $\mathcal{T}_\infty = \varinjlim \mathcal{T}_{(\jmath)}$ of the system $\{\Gamma, f(i,j), i, j = 0, 1, \ldots\}$, where $f(i,j) : \mathcal{T}_{(i)} \to \mathcal{T}_{(j)}, j \geq i$ is a map given by

$$\mathcal{T}_{(j)} = \mathcal{T}_{(i)} \uplus_{h_i} \{\mathcal{F}(v_i), \emptyset\} \uplus \ldots \uplus_{h_{j-1}} \{\mathcal{F}(v_{j-1}), \emptyset\}.$$

Recall that $l(\mathcal{T}_{(\jmath)})$ is the set of all situations associated with the leaf vertices of the event tree $\mathcal{T}_{(\jmath)}$ and let $l_T(\mathcal{T}_{(\jmath)}) \subset l(\mathcal{T}_{(\jmath)})$ be the set of terminating situations associated with the leaf vertices of the event tree $\mathcal{T}_{(\jmath)}$. Define the stages $U_0 = \emptyset$ and $U_1 = \{s_0\}$. For every $\jmath, \jmath = 2, \ldots$, now construct a stage structure as follows:

1. Define the set $\mathscr{I}_\jmath = \{j; \exists e(v_j, v_\jmath) \in \bigcup_t E(t) \cup E_T(t), j < \jmath\}$.
2. Take the set of vectors $R_\jmath = \{\boldsymbol{\rho}; \boldsymbol{\rho} \in \mathcal{L}^{(\jmath)}_{j_1} \times \ldots \times \mathcal{L}^{(\jmath)}_{j_{|\mathscr{I}_\jmath|}}\}$, where $\mathcal{L}^{(\jmath)}_{j_k}$ is the set of categories associated with the variable represented by the vertex $v_{j_k} \in \bigcup_t V(t)$ such that $j_k \in \mathscr{I}_\jmath$.
3. For every $\boldsymbol{\rho} \in R_\jmath$, define a stage

$$U^\jmath_\rho = \{s_i \in l(\mathcal{T}_{(\jmath-1)}) \backslash l_T(\mathcal{T}_{(\jmath-1)}); \Psi_{I_\jmath}(s_i) = \boldsymbol{\rho}\},$$

   where $\Psi_{\mathscr{I}_\jmath}(s_i)$ is a vector $\boldsymbol{\eta} = (\eta_{j_1}, \ldots, \eta_{j_{|\mathscr{I}_\jmath|}})$ such that η_{j_k} is the event associated with the situation $l(\mathcal{T}_{(j_k)})$ along the root-to-s_i path. Also let $U_\jmath = \{U^\jmath_\rho\}_{\rho \in R_\jmath}$.
4. If $l_T(\mathcal{T}_{(\jmath)}) \neq \emptyset$, $U_0 \leftarrow U_0 \cup l_T(\mathcal{T}_{(\jmath)})$.
5. If there is a vertex $v_j, j < \jmath$, such that v_j and $v_\jmath$ represent the same variable $\mathcal{Z}$ whose conditional probability table is time-homogeneous with respect to the time-slices of v_j and $v_\jmath$, $U^j_\rho \leftarrow U^j_\rho \cup U^\jmath_\rho, \rho \in R_\jmath$, and discard $U_\jmath$.

By construction, the DCEG yielded by the event-tree $\mathcal{T}_\infty$ and the stage structure $U = \{U_\jmath, \jmath = 1, 2, \ldots\}$ is a Laminated DCEG and also represents the collection of all conditional independences represented by the corresponding DBN $\mathbb{B}$. ■

Theorem 11. *Every conditional independence statements showed in a 2T-DBN can be expressed in a Two Time-Slice Laminated DCEG.*

Proof. Take a 2T-DBN $\mathbb{B} = (V, E)$. Repeating the construction outlined in the proof of Theorem 10, we obtain the finite staged tree $\mathcal{ST}_1$ corresponding to time-slices t_0 and t_1 of $\mathcal{D}$. Denote by $\mathcal{ST}_0 \subset \mathcal{ST}_1$ the staged tree corresponding to the initial time-slice (t=0) of $\mathcal{D}$. Define the set of finite staged subtrees

$$S(\mathcal{ST}_0) = \{\mathcal{ST}_1(s_i); s_i \in l(\mathcal{ST}_0)\} = \{\mathcal{ST}_{1,j}\}.$$

Also let $\Upsilon(\mathcal{ST}_0) = \{\Upsilon_j(0)\}$ and $\Upsilon_j(0) = \{s_i \in l(\mathcal{ST}_0); \mathcal{ST}(s_i) = \mathcal{ST}_{1,j}\}$. Note that $\emptyset \in \Upsilon$ if a process has a terminating event. Now define the map $h_0 : \Upsilon(\mathcal{ST}_0) \to S(\mathcal{ST}_0)$ such that $h_0(\Upsilon_j) = \mathcal{ST}_{1,j}$. Thus we have that

$$\mathcal{ST}_1 = \mathcal{ST}_0 \uplus_{h_0} S(\mathcal{ST}_0).$$

Since the conditional independent statements entailed in a 2T-DBN $\mathbb{B}$ are time-homogeneous for time-slices $t, t \geq 1$, and only depend on the variable states in time-slices $t-1$ and t, we can write that $\Upsilon(\mathcal{ST}_t) = \{\Upsilon_j(t)\}$, where

$$\Upsilon_j(t) = \{s_i \in l(\mathcal{ST}_t); \xi(s_i, 1) = \xi(s_j, 1), s_j \in l(\mathcal{ST}_1)\}$$

and $h_t = h_0$, for all $t = 2, 3, \ldots$. So $\mathcal{ST}_t = \mathcal{ST}_{t-1} \uplus_{h_{t-1}} S(\mathcal{ST}_0)$ for all $t \geq 2$. Define the set $\Gamma = \{\mathcal{ST}_t; t = 0, 1, \ldots\}$. Now take the staged tree given by the direct limit $\mathcal{ST}_\infty = \varinjlim \mathcal{ST}_k$ of the system $\{\Gamma, f(i,j), i, j \in \mathbb{N}\}$, where f is a morphism such that $\mathcal{ST}_j = \mathcal{ST}_i \uplus_{h_i} S(\mathcal{ST}_i) \ldots \uplus_{h_{j-1}} S(\mathcal{ST}_{j-1})$.

By construction, the staged tree $\mathcal{ST}_\infty$ is a 1 time-homogeneous laminated staged tree spanned by a 0-Periodic Event Tree $(\mathcal{T}_0)$ or 0-Terminated Periodic Event Tree $(\mathcal{T}_0)$, where $\mathcal{T}_0$ is the event tree corresponding to $(\mathcal{ST}_0)$. Adopting the concept of 1-position it then follows that the DCEG supported by $\mathcal{ST}_\infty$ is a Two Time-Slice Laminated DCEG. ■

Example 3 (Dynamic Radicalisation Process - cont.). Figures 2.2 and 6.7 show, respectively, the 2T-DBN and Two Time-Slice Laminated DCEG models corresponding to the radicalisation dynamic described in Example 3. It can be easily verified that every conditional independence statement depicted in the 2T-DBN is also showed in the 2T-DCEG.

However only the symmetric conditional independences exhibited in the Two Time-Slice Laminated DCEG can be graphically read from a 2T-DBN. For instance, take the variable Radicalisation. The context-specific conditional independences associated with this variable are directly depicted in the Two Time-Slice Laminated DCEG. We can see from the graph that the probability of deradicalisation in time $t+1$ given that a prisoner has already adopted radicalisation in time $t, t \geq 1$, (positions w_{19}, w_{20}, w_{21}) is independent of his social contacts in the prison since positions w_{19}, w_{20} and w_{21} are coloured the same (red). This is not so for the 2T-DBN. □

7.3 The Relationship between an NT-DCEG and a CEG

Take a DCEG $\mathbb{C}$ based on a staged tree $\mathcal{ST}_\infty$ and for every time-slice $t, t = 0, 1, \ldots$, construct a CEG $\mathbb{C}_t$ spanned by the staged tree $\mathcal{ST}_t \subset \mathcal{ST}_\infty$. Then for every $t, t = 0, 1, \ldots$, the set of primitive probabilities

$$\Pi_t = \{\pi(s|s_i); s \in ch(s_i), s_i \in \mathcal{T}_t \subset \mathcal{T}_\infty\}$$

defines a consistent probability measure over the path σ-algebra of $\mathbb{C}_t$ (see e.g. Smith and Anderson (2008)), where $\Pi_t \subset \Pi$ and Π is the set of primitive probabilities of $\mathbb{C}$. The path σ-algebras $\mathscr{F}_t = \mathscr{F}(\mathbb{C}_t), t = 0, 1, \ldots$, associated with each CEG $\mathbb{C}_t$ constitute a natural filtration of the path-cylinder σ-algebra $\mathscr{F} = \mathscr{F}(\mathbb{C})$ corresponding to $\mathbb{C}$. The DCEG probability space can then be equipped with a useful set of CEGs $\mathfrak{F}(\mathbb{C}) = \{\mathbb{C}_t; t = 0, 1, \ldots\}$.

Note that in order to build an NT-DCEG $\mathbb{C}$ $-$ and so its corresponding CEGs in $\mathfrak{F}(\mathbb{C})$ $-$, it is necessary to construct only $N + 1$ process-driven objects: the event tree $\mathcal{T}_{-1}$ and the forests $\mathcal{F}o_0, \ldots, \mathcal{F}o_{N-1}$. Remember from Section 7.1 that the graphical topology of each component of all forests $\mathcal{F}o_i, i = 0, \ldots, N-1$, is identical and derives from the same finite tree $\mathcal{T}$. They only differ in the way they are coloured. This coloring represents the different N-Markov probability measure that can be embedded into the event tree described by domain experts. In a 2T-DCEG, it is then sufficient to elicit only the objects $\mathcal{T}_{-1}$, $\mathcal{F}o_0$ and $\mathcal{F}o_1$ since the process is assumed to be 1-Markov time-homogeneous.

Having the same stage structure, both a DCEG $\mathbb{C}$ and a CEG $\mathbb{C}_T$ depict equivalent conditional independences if the interest lies in the 1-step unfolding of events that may happen from a specific situation at time-slice $t, t \leq T$. However, this fact does not hold for analyses that involve a development over two or more steps. This is because positions in a CEG are defined using *finite* subtrees expressing only the early unfoldings of the process whilst positions in a DCEG are based on *infinite* subtrees. Therefore all situations at time $t, t \leq T$, merged into a single position in $\mathbb{C}_T$ will not necessarily be collected by a unique equivalent position in $\mathbb{C}$.

Fortunately there is a stronger link between a NT-DCEG $\mathbb{C}$ and a CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C})$. This enables us to express every $C_t, t = N-1, N, \ldots$, using subgraphs of $\mathbb{C}$. To obtain this result (Theorem 12) we first need to identify these subgraphs and to explain how they can be extracted from $\mathbb{C}$. Before formally introducing this construction, Figure 7.4 depicts schematically how we do this. Observe that every NT-DCEG $\mathbb{C}$ has two important subgraphs, $\mathbb{D}_I$ and $\mathbb{D}_H$. The subgraph $\mathbb{D}_I$ initialises the modelled process over the first $N-1$ time-slices. The cyclic subgraph $\mathbb{D}_H$ represents the time-homogeneous developments of the process and then contains the cyclical temporal edges from time-slice t to $t+1, t = N-1, N, \ldots$. These two subgraphs are connected by a bipartite graph $\mathbb{G}_0$ whose temporal edges I will call transition edges.

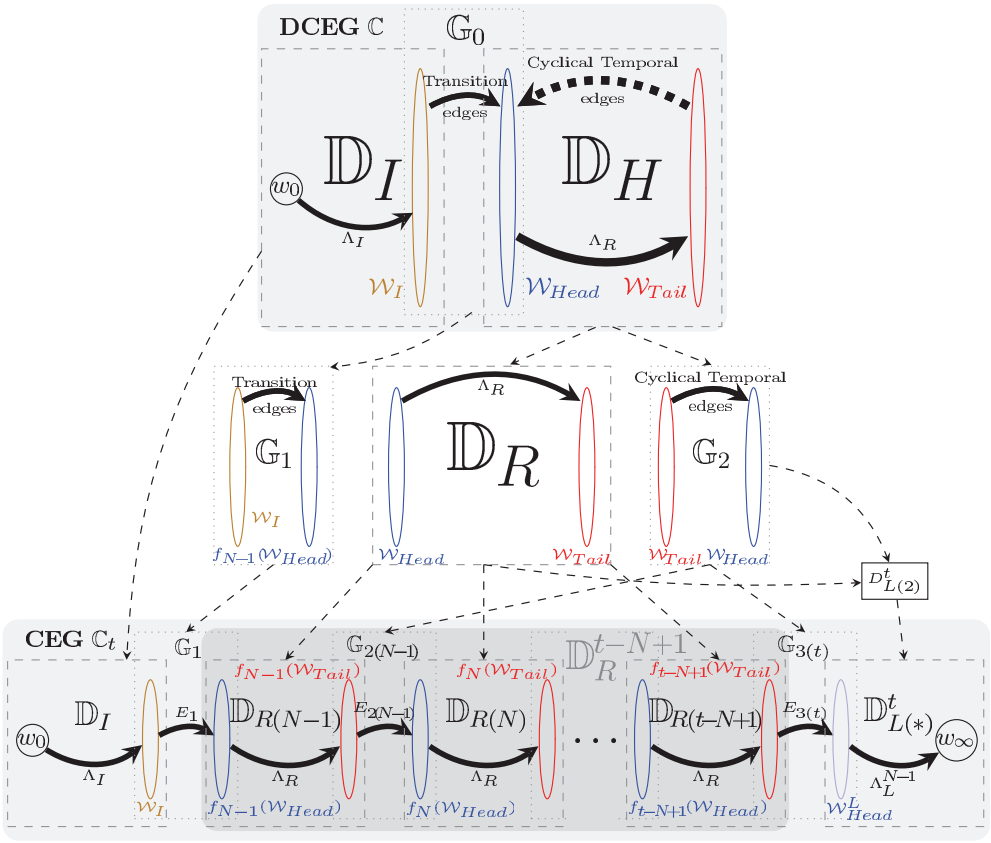


Figure 7.4: The process of obtaining a CEG $\mathbb{C}_t, t \geq 2N-2$, from a DCEG $\mathbb{C}$. Schematic representation of Theorem 12.

Assume that a 2T-DBN $\mathbb{D}$ is re-expressed as a 2T-DCEG $\mathbb{C}$ (Theorem 11). The initial time-slice $\mathbb{D}(0) \subset \mathbb{D}$ and the temporal edges $E_\dagger(1)$ connecting the time-

slices $\mathbb{D}(0)$ and $\mathbb{D}(1)$ of $\mathbb{D}$ correspond, respectively, to the subgraph $\mathbb{D}_I \subset \mathbb{C}$ and the transition edges in $\mathbb{G}_0 \subset \mathbb{C}$. The time-homogeneous time-slices $\mathbb{D}(t)$, $t = 1, 2, \ldots$, and its associated temporal edges $E_\dagger(t), t = 2, 3, \ldots$, are depicted in the subgraph $\mathbb{D}_H \subset \mathbb{C}$. In particular, $\mathbb{D}_H$ yields two subgraphs $\mathbb{D}_R$ and $\mathbb{G}_2$ representing, respectively, the time-slices $\mathbb{D}(t), t = 1, 2, \ldots$, and the temporal edges in $E_\dagger(t), t = 2, 3, \ldots$. Intuitively, Theorem 12 enables us to filtrate the infinite process at a given time t and to unfold the infinite time-homogeneous time-slices summarised in $\mathbb{D}_H$ into a finite CEG $\mathbb{C}_t$ using the graphs $\mathbb{D}_R$ and $\mathbb{G}_2$.

To formally define these subgraphs, let $\mathcal{W}_{Head}$ and $\mathcal{W}_{Tail}$ denote the sets of positions of $\mathbb{C}$ that are, respectively, the heads and tails of cyclical temporal edges. Let $\mathcal{W}_I^t, t = 0, \ldots, N-2$, be the set of all positions in time-slice t that are parents of a position in time-slice $t+1$. The position sets $\mathcal{W}_{Head}$, $\mathcal{W}_{Tail}$ and $\mathcal{W}_I$, where $\mathcal{W}_I = \mathcal{W}_I^{N-2}$, generate the interfaces between the different subgraphs we need to construct. Now take any directed graph $\mathbb{G} = (V, E)$ and let $a(V_a), V_a \subseteq V$, denote the set of all vertices in $\mathbb{G}$ that are antecedents of at least one vertex in V_a.

The initial graph $\mathbb{D}_I = (V_I, E_I) \subset \mathbb{C}$ corresponds to the set of w_0-to-$\mathcal{W}_I$ paths (Λ_I) in $\mathbb{C}$. So formally, we have that

$$V_I = \{w \in \mathbb{C}; w \in a(\mathcal{W}_I) \cup \mathcal{W}_I\} \text{ and } E_I = \{e(w_i, w_j) \in \mathbb{C}; w_i, w_j \in V_I\}.$$

Denote the graph $\mathbb{D}_{I(*)}^t = (V_{I(*)}^t, E_{I(*)}^t) \subset \mathbb{C}$, where

$V_{I(*)}^t = \{w \in \mathbb{C}; w \in a(\mathcal{W}_I^t) \cup \mathcal{W}_I^t \cup ch(\mathcal{W}_I^t)\}$ and $E_{I(*)}^t = \{e(w_i, w_j) \in \mathbb{C}; w_i, w_j \in V_{I(*)}^t\}$.

Finally construct a graph $\mathbb{D}_{I(\infty)}^t$ from each graph $\mathbb{D}_{I(*)}^t$ by merging the set of vertices $\mathcal{W}_{I(*)}^t = \{w \in \mathbb{C}; w \in ch(\mathcal{W}_I^t)\}$ into a single a node and relabelling it as w_∞. When a terminating vertex w_∞ already exists in $V_{I(*)}^t$, it is only necessary to merge the set $\mathcal{W}_{I(*)}^t$ into w_∞.

Define the bijective label transformations $f_t : \mathcal{W} \to \mathcal{W}^t, t = 0, 1, \ldots$, such that: $f_0(w_i) = w_i$; $f_t(w_i) = w_i^t$, for all position $w_i \in \mathcal{W} \backslash \{w_\infty\}$ and $t = 1, 2, \ldots$; and $f_t(w_\infty) = w_\infty$, if $w_\infty \in \mathcal{W}$. Now construct the graph $\mathbb{G}_r = (V_r, E_r), r = 0, 1$, where $V_r = \mathcal{W}_I \cup f_{(N-1)*r}(\mathcal{W}_{Head})$ and

$$E_r = \{e(w_i, f_{(N-1)*r}(w_j)); e(w_i, w_j) \in \mathbb{C}, w_i \in \mathcal{W}_I, w_j \in \mathcal{W}_{Head}\}$$

is the set of edges that goes from an acyclic position at time $N-2, N \geq 2$, to a cyclic position at time $N-1$ in an NT-DCEG $\mathbb{C}$. The isomorphic graphs $\mathbb{G}_r, r = 0, 1$, then provide the link between the time-slices $N-2$ and $N-1$.

Now take the graph $\mathbb{D}_H = (V_H, E_H) \subseteq \mathbb{C} = (V, E)$ made up of the set of $\mathcal{W}_{Head}$-to-$\mathcal{W}_{Tail}$ paths (Λ_R) in $\mathbb{C}$ and the set of cyclical temporal edges $E_{\textcircled{\dagger}} \subseteq E$. Here we then have that $V_H = V \backslash V_I$ and $E_H = E_R \cup E_{\textcircled{\dagger}}$, where

$$E_R = \{e(w_i, w_j) \in E \backslash (E_I \cup E_0 \cup E_{\textcircled{\dagger}})\}.$$

Let $\mathbb{D}_R = (V_R, E_R)$, where $V_R = V_H$, be a subgraph obtained from $\mathbb{D}_H$ when its cyclical temporal edges are removed. A graph $\mathbb{D}_{R(t)} = (V_{R(t)}, E_{R(t)})$ is obtained from $\mathbb{D}_R$ when all positions in V_R are relabelled by the label vertex transformation f_t as follows: $V_{R(t)} = \{f_t(w); w \in V_R\}$ and $E_{R(t)} = \{(f_t(w_a), f_t(w_b)); (w_a, w_b) \in E_R\}$. So, $\mathbb{D}_{R(t)}$ is isomorphic to $\mathbb{D}_R$ and represents $\mathbb{D}_R$ at time t.

To connect together the graphs $D_{R(t)}$, we need to define the graphs

$$\mathbb{G}_{2(t)} = (V_{2(t)}, E_{2(t)}), t = N-1, N, \ldots,$$

where $V_{2(t)} = f_t(\mathcal{W}_{Tail}) \cup f_{t+1}(\mathcal{W}_{Head})$ and $E_{2(t)} = \{e(w_i^t, w_j^{t+1}); e(w_i, w_j) \in E_{\textcircled{\dagger}}\}$. Since the edge set $E_{2(t)}$ is spanned by the set of cyclical temporal edges of $\mathbb{C}$, a graph $\mathbb{G}_{2(t)}$ then represents the dependence structure between time-slices t and $t+1, t = N-1, N, \ldots$. It is also useful to define a particular union operation between two graphs.

Definition 41 (Union Graph)**.** Take two graphs $\mathbb{G}_a = (V_a, E_a)$ and $\mathbb{G}_b = (V_b, E_b)$, where a vertex v with label l_v and an edge $e(v_1, v_2)$ with label l_e are, respectively, defined by a pair (v, l_v) and a triple (v_1, v_2, l_e). A *union graph* of $\mathbb{G}_a = (V_a, E_a)$ and $\mathbb{G}_b = (V_b, E_b)$ is given by $\mathbb{G} = (V, E) = \mathbb{G}_a \oplus \mathbb{G}_b$, where $V = V_a \cup V_b$ and $E_a \cup E_b$.

Now define a graph $\mathbb{D}_R^{t_a, t_b}, t_a < t_b$, using a set of graphs $\mathbb{D}_{R(n)}$ and connective graphs $G_{2(n)}$ using the equation:

$$\mathbb{D}_R^{t_a, t_b} = \mathbb{D}_{R(t_a)} \oplus \mathbb{G}_{2(t_a)} \oplus \mathbb{D}_{R(t_a+1)} \oplus \mathbb{G}_{2(t_a+1)} \oplus \ldots \oplus \mathbb{D}_{R(t_b-1)} \oplus \mathbb{G}_{2(t_b-1)} \oplus \mathbb{D}_{R(t_b)}.$$

Henceforth let $\mathbb{D}_R^{N-1, N-1}$ be equal to $\mathbb{D}_{R(N-1)}$ and denote $\mathbb{D}_R^{N-1, t}$ by $\mathbb{D}_R^t$.

Every time-slice $t, t \geq N-1$, of a CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C}), t \geq N-1$, has a similar stage structure to the one depicted in $\mathbb{D}_R$ because of the time-homogeneity of $\mathbb{C}$. However the number of positions associated with the last κ time-slices can be smaller than the number of positions in $\mathbb{D}_R$, where $\kappa = \min(t-N+2, N-1)$, $t \geq N-1$. This happens because the probabilistic and graphical map identifying two situations by the same vertex in $\mathbb{C}_t$ only holds over a finite tree.

To define a family of graphs representing these last κ time-slices, we next construct the following two set of graphs:

1. $\mathbb{D}^t_{l(a)} = \mathbb{D}^t_R \oplus \mathbb{G}_{2(t)}, t = N-1, \ldots, 2N-3$; and
2. $\mathbb{D}^t_{l(b)} = \mathbb{D}^{t-N+2,t}_R \oplus \mathbb{G}_{2(t)}$, $t = 2N-2, 2N-1, \ldots$.

For $\mathbb{D}^t_{l(i)} = (V^t_{l(i)}, E^t_{l(i)})$, $i = a, b$, the graph $\mathbb{D}^t_{L(i)}$ is then constructed by merging the set of vertices $\mathcal{W}^t_{l(i)} = \{w \in V^t_{l(i)}; w \in f_{t+1}(\mathcal{W}_{Head})\}$ into a single a node and relabelling it as w_∞. If a terminating vertex w_∞ already exists in $V^t_{l(i)}, i = a, b$, it is necessary only to merge the set $\mathcal{W}^t_{l(i)}$ into w_∞.

Finally, to define a vertex contraction operator Φ for a coloured graph, let $\Lambda(v)$ be the set of direct paths that unfolds from a vertex $v \in V$. This operator Φ enables us to introduce the topological simplifications in the set of vertices associated with the last κ time-slices of a CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C})$ that inherits the coloured graphical structure of subgraphs obtained from a DCEG $\mathbb{C}$.

Definition 42 (Vertex Contraction Operator)**.** Take a coloured directed acyclic graph $\mathbb{G} = (V, E)$. The *vertex contraction operator* Φ merges every two vertices $v_a, v_b \in V$ if and only if they are coloured the same and there exists a bijection

$$\phi_v(v_a, v_b) : \Lambda(v_a) \to \Lambda(v_b), \tag{7.3}$$

such that the ordered sequence of edge labels and edge colours in a path $\lambda \in \Lambda(v_a)$ equals the ordered sequence of edge labels and edge colours in the path

$$\lambda' = \phi_v(v_a, v_b)(\lambda) \in \Lambda(v_b).$$

Theorem 12 now asserts that every $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C}), t \geq 2N-2$, can be decomposed in three graphs $\mathbb{D}_I$, $\mathbb{D}^t_R$ and $\mathbb{D}^t_L$; see also Figure 7.5. The graph $\mathbb{D}_I$ corresponds

to the initialisation of our model and the graphs $\mathbb{D}_R^{t-N+1}$ and $\mathbb{D}_L^t$ are associated with the N-Markov time-homogeneity condition.

Theorem 12. *Take an NT-DCEG $\mathbb{C}, N \geq 2$. Then every CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C})$ can be written as*

$$\mathbb{C}_t = \begin{cases} \Phi(\mathbb{D}^t_{I(\infty)}), & \text{if } 0 \leq t \leq N-2, \\ \Phi(\mathbb{D}_I \oplus \mathbb{G}_1 \oplus \mathbb{D}^t_{L(a)}) & \text{if } N-1 \leq t \leq 2N-3, \\ \mathbb{D}_I \oplus \mathbb{G}_1 \oplus \mathbb{D}_R^{t-N+1} \oplus \mathbb{D}^t_L & \text{if } t \geq 2N-2. \end{cases} \tag{7.4}$$

where $\mathbb{D}^t_L = \Phi(\mathbb{G}_{2(t-N+1)} \oplus \mathbb{D}^t_{L(b)})$. If $N=1$, we have that $\mathbb{C}_t = \mathbb{D}_R^{0,t}, t = 0, 1, \ldots$.

Proof. Denote by U_t the stage structure of each time-slice t of an NT-DCEG $\mathbb{C}$. A CEG $\mathbb{C}_T \in \mathfrak{F}(\mathbb{C})$ has the same stage structure U_t for every time-slice $t, t \leq T$. Let $W_t^C, t \geq 0$, and $W_t^{C_T}, t \leq T$, be, respectively, the position structure of $\mathbb{C}$ and $\mathbb{C}_T$ associate with time-slice t. We therefore need to prove that when $T \geq 2N-2$, $W_t^{C_T} = W_t^C$, if $t = 0, \ldots, T-N+1$. If this is true, the result follows by the definition of the subgraphs introduced in this section.

To do this, take two situations s_a and s_b at time-slice $t, t = 0, \ldots, T-N+1$, such that s_a and s_b are, respectively, at two different positions $w_a \in W_t^C$ and $w_b \in W_t^C$ but at the same stage $u_j \in U_t$. Hypothesise now that s_a and s_b are at the same position $w_c \in W_t^{C_T}$. Being at the same position w_c guarantees that there is a graph and probabilistic isomorphism between the stage subtrees $\mathcal{ST}_{t+N-1}(s_a)$ and $\mathcal{ST}_{t+N-1}(s_b)$.

So, we can then map every path $\lambda_a \subset \mathcal{ST}_{t+N-1}(s_a)$ to a path $\lambda_b \subset \mathcal{ST}_{t+N-1}(s_b)$. Because of N-Markov time-homogeneity, there is a probabilistic and graphical isomorphism between the stage trees $\mathcal{ST}(l(\lambda_a))$ and $\mathcal{ST}(l(\lambda_b))$, where $l(\lambda_i), i = a, b$, is a leaf situation of a path $\lambda_i \subset \mathcal{ST}_{t+N-1}(s_i)$. Thus the situations s_a and s_b must be at a same position in the NT-DCEG $\mathbb{C}$ which contradicts our initial hypothesis. So, if two situations at time $t, t = 0, \ldots, T-N+1$, are at different positions in $\mathbb{C}$ then they must be at different positions in $\mathbb{C}_T$.

Observe now that if two situations at time $t, t = 0, \ldots, T-N+1$, are at a same position in $\mathbb{C}$ then they must also be at a same position $\mathbb{C}_T$. This happens because

if the colourful trees unfolding from these two situations are isomorphic then their corresponding finite colourful subtrees associated with $\mathbb{C}_T$ are also isomorphic. This completes the proof. ■

Note that any two subgraphs $\mathbb{D}_{R(t_R)} \subseteq \mathbb{D}_R^{t-N+1}$ and $\mathbb{D}_{L(t_L)} \subseteq \mathbb{D}_L^t$ corresponding, respectively, to time-slices t_R and t_L have the same stage structure but not necessarily isomorphic position structures W_{t_R} and W_{t_L}. However there is a surjection $\varphi : W_{t_R} \to W_{t_L}$, such that $\mathbb{D}_{L(t_L)}$ is obtained from $\mathbb{D}_{R(t_R)}$ by merging all positions in $\varphi^{-1}(w_{t_L}), w_{t_L} \in W_{t_L}$, into a single position w_{t_L}.

This point can be more easily appreciated by re-expressing the last subgraph $\mathbb{D}_L^t$ as

$$\mathbb{D}_L^t = \Phi(\mathbb{G}_{2(t-N+1)} \oplus \mathbb{D}_{L(b)}^t) = \mathbb{G}_{3(t)} \oplus \mathbb{D}_{L(*)}^t, \tag{7.5}$$

where $\mathbb{D}_{L(*)}^t = \Phi(\mathbb{D}_{L(b)}^t)$ and $\mathbb{G}_{3(t)} = (V_{3(t)}, E_{3(t)})$ is obtained from $\mathbb{G}_{2(t-N+1)}$ by vertex contraction operations over the vertices in $f_{t-N+2}(\mathcal{W}_{Head}) \in V_{2(t-N+1)}$ that are merged to form a single vertex in $\mathbb{D}_L^t$. Formally, we then have that

$$V_{3(t)} = f_{t-N+1}(\mathcal{W}_{Tail}) \cup \mathcal{W}_{Head}^L \text{ and } E_{3(t)} = \{e(w_i^{t-N+1}, w_j^{t-N+2}) \in \mathbb{D}_L^t\},$$

where

$$\mathcal{W}_{Head}^L = \{w_i^{t-N+2} \in \mathbb{D}_L^t; w_i^{t-N+2} \in ch(f_{t-N+1}(\mathcal{W}_{Tail}))\}.$$

In Figure 7.5, Λ_L^t denotes the set of $\mathcal{W}_{Head}^L$-to-w_∞ paths in $\mathbb{D}_{L(*)}^t$.

In contrast to a CEG, an NT-DCEG provide us with a very expressive and summary framework for representing conditional statements in a dynamic environment. This is because all the subgraphs $\mathbb{D}_R^t$ and $\mathbb{D}_L^t$ are summarised in the subgraph $\mathbb{D}_H$. Being based on an infinite tree, an NT-DCEG $\mathbb{C}$ also avoids introducing unnecessary refinements of the position structure that its CEGs in $\mathfrak{F}(\mathbb{C})$ might be forced to express. This happens because in a CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C})$ a position corresponds to a set of situations which always share a finite coloured subtree rather than an infinite one. The example below illustrates the concepts discussed in this section.

Example 3 (Dynamic Radicalisation Process - cont.). Recall the 2T-DCEG $\mathbb{C}$ depicted in Figure 6.7. Figure 7.5 shows how to construct the CEG $\mathbb{C}_2 \in \mathfrak{F}(\mathbb{C})$ using the different subgraphs derived from $\mathbb{C}$. As might be expected for a 2T-DCEG,

the graph $\mathbb{D}_I \oplus \mathbb{G}_1 \oplus \mathbb{D}_{R(1)}$ *is similar to* $\mathbb{C}$ *(Figure 6.7) except for the absence of cyclical temporal edges and the addition of superscripts* 1 *to the vertices of* $\mathbb{D}_{R(1)}$.

The graphs $\mathbb{D}_I$ *and* $\mathbb{D}_{R(1)}$ *are based, respectively, on the event tree* $\mathcal{T}_0$ *and the forest* $\mathcal{F}o_1 = \{\mathcal{T}_1(s_{13+2i}); i = 0, \ldots, 8\}$ *(see Figure 6.4). The graph* $\mathbb{D}_R$ *is topologically identical to* $\mathbb{D}_{R(1)}$ *but with the vertex superscript* 1 *removed. The connective subgraph* $\mathbb{G}_1$ *is defined by the set of positions* $V_1 = \{w_4, \ldots, w_9, w^1_{10}, \ldots, w^1_{15}\}$ *and the set of transition edges* $E_1 = \{(w_4, w^1_{10}), \ldots, (w_9, w^1_{15})\}$.

The subgraph $\mathbb{D}^1_R$ *is made up of only one repetition,* $\mathbb{D}_{R(1)}$*, of the subgraph* $\mathbb{D}_R$. *So no bipartite graph* $\mathbb{G}_{2(t)}$ *needs to be depicted in Figure 7.5. The subgraph* $\mathbb{D}^2_L$ *can be directly obtained from* $\mathbb{D}_R$ *by relabelling its vertices and by merging the set of positions* $\{w_{19}, w_{20}, w_{21}\}$, $\{w_{22}, w_{23}, w_{24}\}$ *and* $\{w_{25}, w_{26}, w_{27}\}$ *of* $\mathbb{D}_R$ *into, respectively, the positions* w^2_{19}, w^2_{22} *and* w^2_{25} *of* $\mathbb{D}^2_L$. *These positions are gathered into* $\mathbb{D}^2_L$ *because they have isomorphic unfolding developments over a single transition of a time-slice.*

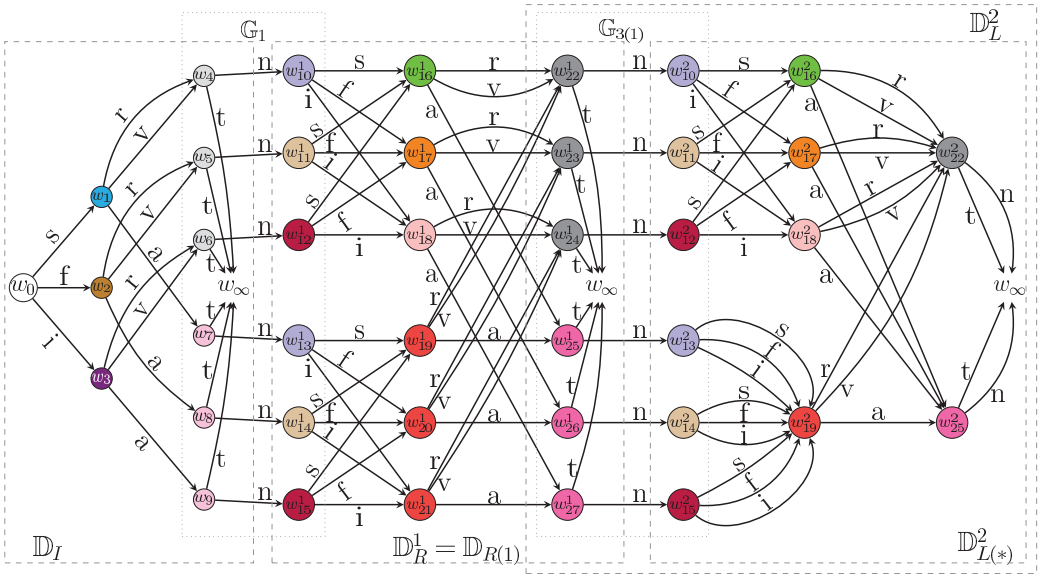


Figure 7.5: The CEG $\mathbb{C}_2$ associated with the 2T-DCEG depicted in Figure 6.7

□

7.4 Reading Conditional Independences

Conditional independence statements implicit in a NT-DCEG $\mathbb{C}$ at time-slice T can be read from a CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C}), t \geq T$, because the supporting staged trees of both graphical models are identical from time 0 to T. So fortunately we can read the conditional independences using the well-developed CEG techniques (Smith and Anderson, 2008, Smith, 2010, Thwaites and Smith, 2011). However, for large T, to apply this correspondence directly can be time-consuming and laborious. Theorem 12 allows us to circumvent these inefficiencies by using the DCEG topology itself to examine its implicit conditional independences.

Assume that our focus is on reading the existing conditional independences at time t given that all past events are known. It is obvious that the conditional independences embedded into $\mathbb{C}$ at time-slices $t, t = 0, \ldots, N-1$, can be directly read from its subgraph $\mathbb{D}_I$. Theorem 13 then guarantees that the conditional independences at a time $t, t = N-1, \ldots$, can also be directly interpreted using $\mathbb{C}$. This is possible for two reasons. First, since every process represented by a NT-DCEG is N-Markov, we only need information over the last $N-1$ time-slices. This is completely represented in $\mathbb{D}_I$. Second, Theorem 12 assures us that the conditional independences at a time $t, t = N-1, \ldots$, are depicted by a subgraph $\mathbb{D}_{R(t)}$ that is isomorphic to a subgraph $\mathbb{D}_R$.

Let $w(t)$ be a position of a DCEG/CEG associated with a time-slice t. Also let $\Xi_c(w(t), N) = \{\xi(s, N); s \in w(t)\}$ denote a set of all sequences of events $\xi(s, N), s \in w(t)$, that happen along each walk from the root position w_0 to $w(t)$ whose events from time 0 to $t - N$ are excluded.

Theorem 13. *Take an NT-DCEG $\mathbb{C} = (V, E)$ and define the set of positions $\mathcal{W}_{Head} \subseteq V$ according to Definition 39. In a CEG $\mathbb{C}_T = (V_T, E_T) \in \mathfrak{F}(\mathbb{C})$, $T \geq 2N-2$, for every position $w_a^*(t) \in f_t(\mathcal{W}_{Head}) \subseteq V_T, N \leq t \leq T-N+1$, we have that*

$$\Xi_c(w_a^*(t), N-1) = \Xi_c(w_b^*(N-1), N-1) \tag{7.6}$$

where $w_b^*(N-1) = f_{N-1}(f_t^{-1}(w_a^*(t))) \in V_T$.

Proof. Suppose that this result does not hold. Then, according to Theorem 12 $\mathbb{C}_T$ contains at least one position $w_a^*(t) \in f_t(\mathcal{W}_{Head})$, $N \leq t \leq T-N+1$, such that

$$\Xi_c(w_a^*(t), N-1) - \Xi_c(w_b^*(N-1), N-1) = \Xi_\delta = \{\xi_i(w_a^*(t), N-1)\} \neq \emptyset$$

and $w_b^*(N-1) = f_{N-1}(f_t^{-1}(w_a^*(t)))$. Note that in the DCEG $\mathbb{C}$ the positions $w_a = f_t^{-1}(w_a^*(t))$ and $w_b = f_{N-1}^{-1}(w_b^*(N-1))$ are the same: $w_a \equiv w_b$.

For every sequence of events $\xi_i \in \Xi_\delta$ there is therefore a position $w_c^*(N-1) \in V_T$, such that $\xi_i \in \Xi_c(w_c^*(N-1), N-1)$. So, in the event tree associated with $\mathbb{C}$ there are at least two situations s_a and s_c, such that $s_a \in w_a$, $s_c \in w_c = f_{N-1}^{-1}(w_c^*(N-1))$, and both situations descend from the same leaf node of $\mathcal{T}_{-1}$ if $\mathcal{T}_{-1} \neq \emptyset$. Note that $w_c \in V$.

Because of the strong 1-periodicity and time-homogeneity after time $N-1$, it follows that there is an isomorphism between the staged subtrees that unfold from situations s_a and s_c. Therefore, both situations are in the same position. Thus we have that $w_a \equiv w_c$ in $\mathbb{C}$. This implies that $w_b \equiv w_c$ and so

$$w_b^*(N-1) \equiv w_c^*(N-1).$$

This means that $\{\xi_i(w(t), N-1)\}$ is empty. The result then follows by contradiction. ■

Example 9 (Extended Dynamic Radicalisation Process - cont.). Now recall the NT-DCEG depicted in Figure 7.2. Under this model we can directly read that the probability of prison transferring at the initial time is only affected by the prisoners' previous convection and their radicalisation level. As discussed above, to read the conditional independences from time 1 on we can discard the temporal edges and use the initial time-slice as a supporting legend. For example, note that at any time $t, t \geq 1$, the chance of deradicalising a radical prisoner who hasn't be transferred does not depend on his social network in the prison. However the inmate's criminal background can have an impact on this process. This might suggest that an isolation policy is not been the best way to handle radical prisoners and that decision makers should consider the prisoners' previous convictions when designing their policies. □

The analysis of how a process can unfold s steps ahead from time t given a particular set of past events $\mathcal{E}$ needs a little more care because of the cyclical temporal edges. Observe that the set $\mathcal{E}$ corresponds to a set of positions $\mathcal{W}_{\mathcal{E}}$ at the beginning of time t, i.e. every position in $\mathcal{W}_{\mathcal{E}}$ has at least one parent in time $t-1$. If interest is in a time-slice $t+s$ that happens within the first $N-1$ time-slices ($t+s \leq N-1$), the analysis using an NT-DCEG is simplified by discarding the walks that do not unfold from $\mathcal{W}_{\mathcal{E}}$. The same procedure also applies if the focus is only at one time-slice ($s = 1$) ahead from the present time t, $t = N-1, N, \ldots$. In this case, we have that $\mathcal{W}_{\mathcal{E}} \subseteq \mathcal{W}_{Head}$.

Example 3 (Dynamic Radicalisation Process - cont.). Assume that in his most recent period t in prison an inmate adopting radicalisation kept intense social contacts with extremist recruiters. We are concerned about what might happen to him at the next time step were he to stay in the same prison. In this case, we have that the set of past events $\mathcal{E}$ corresponds to the set of events

$$E(t) = \{(N(t) = i, R(t) = a, T(t) = n)\}$$

at time-slice t, and our focus is on the developments that might happen at time-slice $t+1$.

Using the 2T-DCEG elicited in Figure 6.7 as representative of this process, the possible future developments associated with the event set $\mathcal{E}$, where $\mathcal{W}_{\mathcal{E}} = \{w_{15}\}$, is highlighted in Figure 7.6 below. Two points stand out. First, our model implies that observing the social contacts of the target inmate at time $t+1$ will not provide any additional information about his development within this time interval useful to answer the question above.

Second, in the end of the time-slice $t+1$ the prisoner adopting radicalisation could arrive at any possible position in the set $\{w_\infty, w_i; i=10, \ldots, 15\}$. This indicates that the prison managers might lose track of him. For example, this prisoner could intentionally reshape his social networks at time-slice $t+1$ to disguise his extremist ideology. The consequence would be that at the beginning of time-slice $t+2$ he could be at position w_{13}. Note that inmates at positions w_{10} and w_{13} have the

same social pattern.

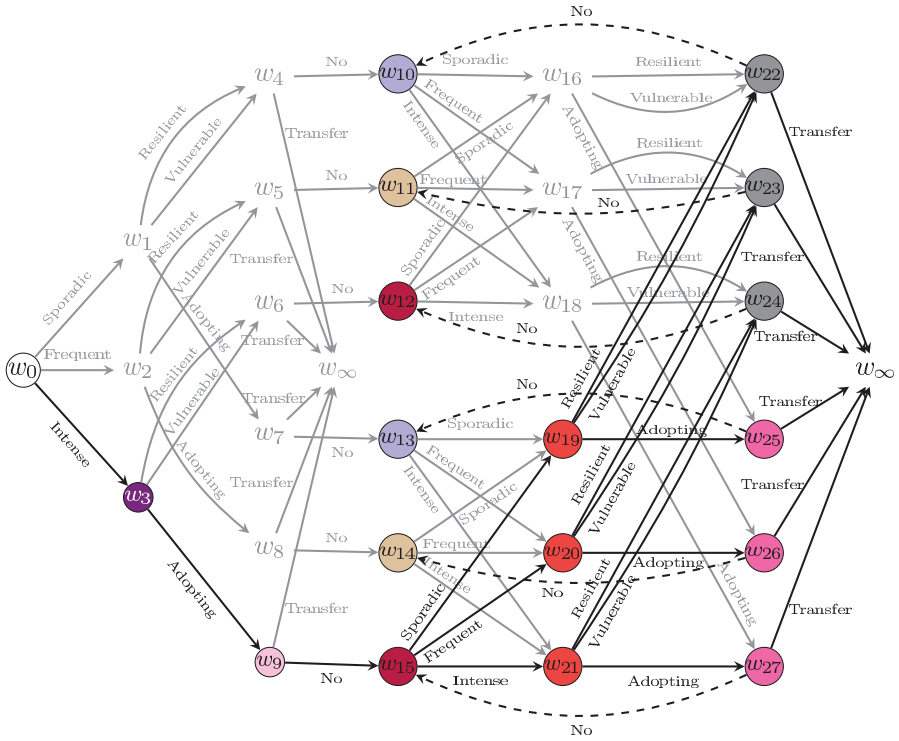


Figure 7.6: The 2T-DCEG associated with Example 3 when a radical prisoner maintains intense social contacts with other radical inmates and is not transferred.

Moreover, classifying an inmate as adopting radicalisation is a challenging task and error prone. At time $t+2$ these facts might misguidedly prompt prison managers to downgrade the inmate's classification as adopting radicalisation given at time t and so reduce the monitoring mechanisms over him. So the reasoning the 2T-DCEG provokes is useful: it might stimulate some pro-active response immediately to deradicalise the inmate at time $t+1$ or at least to monitor him closely for a long time. □

When s and $t+s$ are, respectively, greater than 1 and N, to explore how a process might unfold over s time steps after the actual time t given a particular set of past events $\mathcal{E}$ needs more attention. However the task can be easily simplified if the transition matrix $\boldsymbol{M}$ associated with the Markov Chain projection of the elicited NT-DCEG (Section 7.1) is used.

This assumes that t is greater than $N-2$. We are then able to identify from $\mathcal{E}$

which positions $w \in \mathcal{W}_{Head}$ a unit may be in at time $t+s$. Let $\boldsymbol{w}(t)$ and $p(\boldsymbol{w}(t))$ be, respectively, a binary vector that represents this location information and its corresponding probability vector at time t. Then

$$p(\boldsymbol{w}(t+s)) = p(\boldsymbol{w}(t)) \times \boldsymbol{M}^{s-1}. \tag{7.7}$$

Now based on the vector $p(\boldsymbol{w}(t+s))$ we can define $\mathcal{W}_{\mathcal{E}} \subseteq \mathcal{W}_{Head}$ and then use the same framework described for $s=1$ to analyse what might happen at time-slice $t+s$. Note that when $t \leq N-2$ and $t+s \geq N-1$, before applying Equation 7.7 we first need to project our current information at time t into the future time-slice $N-1$. We therefore need to use the transitions depicted in the initial subgraph $\mathbb{D}_I$ in order to find the set of positions that a unit can be at time-slice $N-1$ based on its possible positions at time t. In this case, it follows that

$$p(\boldsymbol{w}(t+s)) = p(\boldsymbol{w}(t)) \times \boldsymbol{M}_t \times \boldsymbol{M}^{T+s-N}, \tag{7.8}$$

where $\boldsymbol{M}_t, t \leq N-2$, is a transition matrix associated with the positions in $\mathbb{D}_I$ from time-slice t to time-slice $N-1$.

7.5 Local independence and Granger noncausality

Schweder (1970) first introduced the concept of local independence for Markov processes. Subsequently Aalen (1987) generalised this and applied it to processes with a Doob-Meyer decomposition. Didelez (2008) then further extended the concept so that it applied to general multivariate processes. The notion of local independence is useful because in a model that fully represents a process a local independence statement can be translated into Granger noncausality (Granger, 1969); see e.g. (Eichler, 2007) and (Didelez, 2008). Granger noncausality has recently also been discussed for mediation and intervention (Eichler and Didelez, 2010, Aalen et al., 2012). Here I develop the idea of local independence for the discrete time processes expressed within a DCEG. For technical consistency, if there is a terminating event in a DCEG, the following concepts of conditional

independences are valid for a unit that experiences a terminating event at time T as long as $t = 0, \ldots, T-1$.

Consider two random variables X and Y that take value over each time-slice. By saying that X is locally independent from Y we mean that the past values of Y do not provide any additional information to predict the current value of X given all set of past events up to the current time. Note that in a DCEG model these variables do not need to begin happening at the initial time-slice $t = 0$; they can start to happen later. Also observe that in some DCEGs we may be interested only in time-slices from a certain time T on. This is often the case for an NT-DCEG model where the experts tend to focus on its time-homogeneous subgraph $\mathbb{D}_R$.

To handle these cases, building on previous work by Eichler (2007), Didelez (2008) and Eichler and Didelez (2010) I introduce the concept of T-local independence below. This enables us to analyse the impact of past events on the current target process from a time-slice T on. This idea directly generalises to random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$. Let $\mathcal{E}^{(t)}$ denote the collection of all sequences of events that happened up to the end of time-slice t and let $\mathcal{E}^{(t)}_{(-\boldsymbol{X})} \subset \mathcal{E}^{(t)}$ denote the history of past events that excludes information with respect a random vector $\boldsymbol{X}$.

Definition 43 (Local Independence)**.** Take two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ measurable with respect every time-slice $t, t \geq T$, of a DCEG. A vector $\boldsymbol{X}$ is said to be T-*locally independent* from $\boldsymbol{Y}$ if all probability distributions $p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)})$ are measurable with respect to $\mathcal{E}^{(t-1)}_{(-\boldsymbol{Y})}$ for all $t = T, T+1, \ldots$. Denote this by $\boldsymbol{X} \underset{\Rightarrow}{\perp\!\!\!\perp}_T \boldsymbol{Y}$. If the local independence condition holds for all time-slice $t, t \geq 0$, then $\boldsymbol{X}$ is said to be *locally independent* from $\boldsymbol{Y}$. Henceforth I will denote local independence by $\boldsymbol{X} \underset{\Rightarrow}{\perp\!\!\!\perp} \boldsymbol{Y}$.

Assuming that the underlying event tree of a DCEG $\mathbb{C}$ completely describes the natural behaviour of a process, a random vector $\boldsymbol{Y}$ is (strongly) Granger noncausal for $\boldsymbol{X}$ with respect to $\mathbb{C}$ if $\boldsymbol{X} \underset{\Rightarrow}{\perp\!\!\!\perp} \boldsymbol{Y}$. Otherwise, we say that $\boldsymbol{Y}$ is Granger causal or a *prima facie* cause for $\boldsymbol{X}$. Analogously, we say that a random vector $\boldsymbol{Y}$ is T-Granger noncausal for $\boldsymbol{X}$ with respect to $\mathbb{C}$ if $\boldsymbol{X} \underset{\Rightarrow}{\perp\!\!\!\perp}_T \boldsymbol{Y}$. For the validity of the

Granger causal interpretation, events not depicted in the event tree cannot span Granger causal structures between random variables measurable with respect the corresponding DCEG.

The T-local independence relation is not necessarily symmetric and therefore neither is Granger noncausality (Didelez, 2008). For example, in the 2T-DCEG depicted in Figure 6.7 the network variable N is locally independent from the radicalisation variable R but the inverse relation does not hold. So under the assumption that this model is a fair representation of the radicalisation process in a prison we can say that R is Granger noncausal for N whilst N is a *prima facie* cause for R.

In discrete time we are often also interested in exploring intra-time conditional independences that characterize each time-slice given the whole set of past events. This differentiates the DCEG from a continuous time graphical model (Gottard, 2007, Didelez, 2008) where two different counting processes cannot represent the same event. In those frameworks, a prisoner is assumed not to radicalise and to change his social network at the same time.

In this respect the DCEG models come closest to the path diagrams used to visualise the dynamic of multivariate weakly stationary multivariate time series (Eichler, 2007). However path diagrams have a different graphical semantic from DCEG models because there vertices represent processes, directed edges correspond to local dependences and dashed edges depict intra-time dependences. This makes them unable to represent any graphically context-specific hypotheses. In these models all time-slices also have the same conditional independence structure.

Definition 44 (Contemporaneous Independence)**.** Take two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ measurable with respect to every time-slice $t, t = T, T+1, \ldots$, of a DCEG. These variables are said to be T-*contemporaneously independent* if for every t, $t{=}T, T{+}1, \ldots$, their joint probability distribution is such that

$$p_{\boldsymbol{X}(t),\boldsymbol{Y}(t)}(\boldsymbol{x}(t), \boldsymbol{y}(t)|\mathcal{E}^{(t-1)}) = p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)}) p_{\boldsymbol{Y}(t)}(\boldsymbol{y}(t)|\mathcal{E}^{(t-1)}). \qquad (7.9)$$

This will be denoted by $\boldsymbol{X} \overset{\perp\!\!\!\perp_T}{\sim} \boldsymbol{Y}$. If this property holds for all time-slices, the variables are simply said to be *contemporaneously independent* and the subscript T

can be dropped from the notation.

For the purpose of this thesis it is therefore useful to introduce the concept of T-contemporaneous independence; see Definition 44. Here I follow some previous authors (Granger, 1980, Eichler, 2007, Eichler and Didelez, 2010). Finally the stochastic independence given in Definition 45 below establishes the condition for two random vectors to be globally independent given the past events. Theorem 14 guarantees that this kind of stochastic independence only happens in the presence of contemporaneous and local independences. This result provides us with a framework that enables us to determine whether or not two variables are stochastically independent without verifying the validate of Equation 7.10 using various algebraic calculations. For example, from Figure 6.7 we can read directly from this 2T-DCEG that the variables N, R and T are not stochastically independent since they are not contemporaneously independent.

Definition 45 (Stochastic Independence)**.** Take two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ measurable with respect every time-slice $t, t \geq T$, of a DCEG. These variables are *T-stochastically independent* if for every $t, t = T, T+1, \ldots$, their joint probability distribution is such that

$$p_{\boldsymbol{X}(t),\boldsymbol{Y}(t)}(\boldsymbol{x}(t), \boldsymbol{y}(t)|\mathcal{E}^{(t-1)}) = p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)}_{(-\boldsymbol{Y})})p_{\boldsymbol{Y}(t)}(\boldsymbol{y}(t)|\mathcal{E}^{(t-1)}_{(-\boldsymbol{X})}). \quad (7.10)$$

This is denoted by $\boldsymbol{X} \perp\!\!\!\perp_T \boldsymbol{Y}$. If this property holds for all time-slices, the variables are said to be *stochastically independent*. We then simply denote this by writing $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}$.

Theorem 14. *Two random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ measurable with respect a DCEG are T-stochastically independent if and only if they are mutually T-locally independent and T-contemporaneously independent.*

Proof. Let $\boldsymbol{X}^{(t)} = (\boldsymbol{X}(0), \boldsymbol{X}(1), \ldots, \boldsymbol{X}(t))$. Assuming that the random vectors are T- stochastically independent, it then follows from Equation 7.10 that for every t, $t = T, T+1, \ldots$,

$$
\begin{aligned}
p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)}) &= \sum_{\boldsymbol{y}(t)} p_{\boldsymbol{X}(t),\boldsymbol{Y}(t)}(\boldsymbol{x}(t),\boldsymbol{y}(t)|\mathcal{E}^{(t-1)}) \\
&= \sum_{\boldsymbol{y}(t)} p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)}_{(-\boldsymbol{Y})}) p_{\boldsymbol{Y}(t)}(\boldsymbol{y}(t)|\mathcal{E}^{(t-1)}_{(-\boldsymbol{X})}) \\
&= p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)}_{(-\boldsymbol{Y})}).
\end{aligned}
$$

Of course, we can obtain a completely analogous result for $\boldsymbol{Y}^{(t)}$. So these vectors are mutually T-locally independent. Substituting this result into Equation 7.10 it is straightforward to see that these vectors are also T-contemporaneously independent.

Conversely it is also true that

$$
\begin{aligned}
p_{\boldsymbol{X}(t),\boldsymbol{Y}(t)}(\boldsymbol{x}(t),\boldsymbol{y}(t)|\mathcal{E}^{(t-1)}) &= p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)}) p_{\boldsymbol{Y}(t)}(\boldsymbol{y}(t)|\mathcal{E}^{(t-1)}) \\
&= p_{\boldsymbol{X}(t)}(\boldsymbol{x}(t)|\mathcal{E}^{(t-1)}_{(-\boldsymbol{Y})}) p_{\boldsymbol{Y}(t)}(\boldsymbol{y}(t)|\mathcal{E}^{(t-1)}_{(-\boldsymbol{X})}).
\end{aligned}
$$

Note that the first and second equalities follows, respectively, from the assumptions that the vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ are T-contemporaneously independent and mutually T-locally independent. ■

7.6 Constructing random variables

Sometimes a Laminated DCEG $\mathbb{C}$ can also be described by a context-specific DBN. In this case a useful class of random variables is one taking its levels as positions that are equally distant from the root position in $\mathbb{C}$. However especially when its tree is asymmetric such random variables are not the only or even the most important class of random variables that can be constructed from an DCEG.

In this section I will present two constructions of random variables intrinsically associated with an NT-DCEG $\mathbb{C}$=(V, E). I will then show how particularly useful conditional independences can be defined between them. This will first require us to extend the concepts of cut and fine cut from a CEG (Smith and Anderson, 2008) so that they can be interpreted analogously in an NT-DCEG.

For an NT-DCEG $\mathbb{C}$, define the following finite set of positions associated with the temporal edges from time $t, t = -1, \ldots, N-2$, to time $t+1$:

$$\mathcal{W}^{t}_{Tail} = \{w(t); (w(t), w_a(t+1)) \in E_{\dagger}, \text{ for some } w_a(t+1)\} \text{ and}$$

$$\mathcal{W}^{t+1}_{Head} = \{w(t+1); (w_a(t), w(t+1)) \in E_{\dagger}, \text{ for some } w_a(t)\}.$$

For $t = N-1, N, \ldots$, fix $\mathcal{W}^{t}_{Tail} = \mathcal{W}_{Tail}$ and $\mathcal{W}^{t}_{Head} = \mathcal{W}_{Head}$. Take the graph $\mathbb{C}^{-}=(V^{-}, E^{-})$, where $V^{-}=V$ and $E^{-}=E-E_{\textcircled{\dagger}}$, and let $\mathcal{W}^{t(*)}_{Tail} = \mathcal{W}^{t}_{Tail} \cup \{w_{\infty}\}$. Now let Λ^{cut}_{t}, $t = 0, 1, \ldots$, denote the set of all w_0-to-$\mathcal{W}^{t}_{Head}$-to-$\mathcal{W}^{t(*)}_{Tail}$ paths in $\mathbb{C}^{-}$. For $t = -1$, when $\mathcal{T}_{-1} \neq \emptyset$, and for $t = 0$, when $\mathcal{T}_{-1} = \emptyset$, let Λ^{cut}_{t} be the set of all w_0-to-$\mathcal{W}^{t(*)}_{Tail}$ paths in $\mathbb{C}^{-}$.

Definition 46 (Cut)**.** In an NT-DCEG $\mathbb{C}$, a *cut* $\mathcal{U}^{cut}_{t}$, $t = -1, 0, \ldots$, is a set of stages such as all paths in Λ^{cut}_{t} pass through exactly one position $w(t) \in u$, for some $u \in \mathcal{U}^{cut}_{t}$. A cut $\mathcal{U}^{cut}_{-1}$ only exists if $\mathcal{T}_{-1} \neq \emptyset$. Let $\mathcal{U}^{cut}$ denote any of the identical cuts $\mathcal{U}^{cut}_{t}$, $t = N-1, N, \ldots$.

Definition 47 (Fine Cut)**.** In an NT-DCEG $\mathbb{C}$, a *fine cut* $\mathcal{W}^{cut}_{t}$, $t = -1, 0, \ldots$, is a set of positions $w(t)$ such that all paths in Λ^{cut}_{t} pass through exactly one position $w(t) \in \mathcal{W}^{cut}_{t}$. A fine cut $\mathcal{W}^{cut}_{-1}$ only exists if $\mathcal{T}_{-1} \neq \emptyset$. Let $\mathcal{W}^{cut}$ denote any of the identical cuts $\mathcal{W}^{cut}_{t}$, $t = N-1, N, \ldots$.

Define a function a h such as $h(x) = 0$, if $x \leq N-1$, and $h(x) = x - N + 1$, otherwise. Let $\Lambda(\mathcal{U}^{cut}_{t}) = \cup_{u \in \mathcal{U}^{cut}_{t}} \Lambda(u, t)$, where $\Lambda(u, t)$ denotes the set of all walks $\lambda = (w_0, \ldots, w(t)) \subset \mathbb{C}$, such that $w(t) \in u$ and each walk λ passes through cycle temporal edges exactly $h(t)$ times. Also let $\mathcal{E}(u)$ and $\mathcal{E}(\mathcal{U}^{cut}_{t})$ denote the set of events that can happen immediately after a unit arriving, respectively, at a particular stage u and at any stage in a cut $\mathcal{U}^{cut}_{t}$. I can now introduce three useful random variables that can be constructed from the cut $\mathcal{U}^{cut}$ taking values over time-slices $t, t = -1, 0, \ldots$. These are defined as follows:

1. $X(\mathcal{U}^{cut}_{t})$ is defined to be a downstream random variable whose state space is the set $\mathbb{X}(\mathcal{U}^{cut}_{t}) = \{1, 2, \ldots, |\mathcal{E}(\mathcal{U}^{cut}_{t})|\}$, such that there exists a bijection

$$\zeta_{X(\mathcal{U}^{cut}_{t})} : \mathbb{X}(\mathcal{U}^{cut}_{t}) \to \mathcal{E}(\mathcal{U}^{cut}_{t}).$$

   Its probability mass function $\pi_X(x)$ is given by

$$\pi_X(x) \propto \sum_{\lambda \in \Lambda_x(\mathcal{U}_t^{cut})} \pi(w'(l(\lambda)) = \zeta_X(x)|l(\lambda)) \prod_{\substack{w \in \lambda \\ w \neq l(\lambda)}} \pi(w'(w)|w),\ x \in \mathbb{X}(\mathcal{U}_t^{cut}), \tag{7.11}$$

where $w'(w)$ is the successor of w in λ, $l(\lambda)$ is the last position of a directed walk λ, and $\Lambda_x(\mathcal{U}_t^{cut})$ is the set of all walks $\lambda \in \Lambda(\mathcal{U}_t^{cut})$, such that the event $\zeta_{X(\mathcal{U}_t^{cut})}(x)$ can unfold from λ in $\mathbb{C}$.

2. $Q(\mathcal{U}_t^{cut})$ is defined to be a separator random variable whose state space is given by the set $\mathbb{Q}(\mathcal{U}_t^{cut}) = \{1, 2, \ldots, |\mathcal{U}_t^{cut}|\}$, such that there is a bijection

$$\zeta_{Q(\mathcal{U}_t^{cut})} : \mathbb{Q}(\mathcal{U}_t^{cut}) \to \mathcal{U}_t^{cut}.$$

The probability mass function $\pi_Q(q)$ is proportional to the sum of all the monomials in primitives associated with $\lambda \in \Lambda(u, t)$, where $u = \zeta_{Q(\mathcal{U}_t^{cut})}(q)$. So explicitly we have that

$$\pi_Q(q) \propto \sum_{\lambda \in \Lambda(\zeta_Q(q), t)} \prod_{\substack{w \in \lambda \\ w \neq l(\lambda)}} \pi(w'(w)|w), \quad q \in \mathbb{Q}(\mathcal{U}_t^{cut}). \tag{7.12}$$

3. $Z(\mathcal{U}_t^{cut})$ is the upstream random variable of $\mathcal{U}_t^{cut}$ in $\mathbb{C}$ whose state space is defined by the set $\mathbb{Z}(\mathcal{U}_t^{cut}) = \{1, 2, \ldots, |\Lambda(\mathcal{U}_t^{cut})|\}$, such that there is a bijection

$$\zeta_{Z(\mathcal{U}_t^{cut})} : \mathbb{Z}(\mathcal{U}_t^{cut}) \to \Lambda(\mathcal{U}_t^{cut}).$$

Its probability mass function $\pi_Z(z)$ is given by

$$\pi_Z(z) \propto \prod_{\substack{w \in \lambda \\ w \neq l(\lambda)}} \pi(w'(w)|w), \quad z \in \mathbb{Z}(\mathcal{U}_t^{cut}), \tag{7.13}$$

where $\lambda = \zeta_{Z(\mathcal{U}_t^{cut})}(z)$.

Note that '$=$' can replace '$\propto$' in the three equations above if the NT-DCEG does not have a sink position w_∞. Let $\mathbb{X}(u) = \{1, 2, \ldots, |\mathcal{E}(u)|\}$ be the state space of the usual random variable $X(u)$ associated with a stage u. From the constructions above we can now immediately recover each random variable $X(u)$, $u \in \mathcal{U}_t^{cut}$, as follows

$$\pi(X(\mathcal{U}_t^{cut}) = x|Q(\mathcal{U}_t^{cut}) = q) = \begin{cases} \pi(X(u_q) = x|u_q) & \text{if } \zeta_{X(\mathcal{U}_t^{cut})}(x) \in \mathcal{E}(u_q), \\ 0 & \text{if } \zeta_{X(\mathcal{U}_t^{cut})}(x) \notin \mathcal{E}(u_q), \end{cases} \tag{7.14}$$

where $u_q = \zeta_{Q(\mathcal{U}_t^{cut})}(q)$.

Theorem 15 below tells us that the actual state of a process given by a stage u determines its immediate development regardless of the possible unfolding walk taken by a unit from the root position to a position $w \in u$. Furthermore, Equation 7.16 guarantees that this is the only conditional independence statement that can be read between an downstream and upstream variables $X(\mathcal{U}_t^{cut})$ and $Z(\mathcal{U}_t^{cut})$ measurable with respect to an NT-DCEG.

Theorem 15. *Take a cut $\mathcal{U}_t^{cut}$, $t = -1, 0, 1, \ldots$, in an NT-DCEG $\mathbb{C}$. Then*

$$X(\mathcal{U}_t^{cut}) \perp\!\!\!\perp Z(\mathcal{U}_t^{cut}) | Q(\mathcal{U}_t^{cut}). \tag{7.15}$$

Additionally, if a function $f(Z(\mathcal{U}_t^{cut}))$ satisfies

$$X(\mathcal{U}_t^{cut}) \perp\!\!\!\perp Z(\mathcal{U}_t^{cut}) | f(Z(\mathcal{U}_t^{cut})), \tag{7.16}$$

then $Q(\mathcal{U}_t^{cut})$ is a function of $f(Z(\mathcal{U}_t^{cut}))$ with probability one. These results also hold when a cut $\mathcal{U}_T^{cut}$ is defined in a CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C})$, $t = T, T+1, \ldots$.

Proof. By definition, if the value of $Q(\mathcal{U}_t^{cut})$ is observed, for example q, then any random variable based on a stage $\zeta_{Q(\mathcal{U}_t^{cut})}(q)$ is completely defined. So none of the w_0-to-$w(t)$ walks, $w(t) \in \zeta_{Q(\mathcal{U}_t^{cut})}(q)$, can bring any additional information on the realization of $X(\mathcal{U}_t^{cut})$. Thus, $X(\mathcal{U}_t^{cut}) \perp\!\!\!\perp Q(\mathcal{U}_t^{cut}) | Z(\mathcal{U}_t^{cut})$.

Assume that $Q(\mathcal{U}_t^{cut})$ is not a function of $f(Z(\mathcal{U}_t^{cut}))$ with probability one. Then there are at least two non-zero probability walks λ_1 and λ_2 in $\Lambda(\mathcal{U}_t^{cut})$ such that $l(\lambda_1)$ and $l(\lambda_2)$ are in two different stages, $f(z_1) = f(z_2)$ and

$$X(\mathcal{U}_t^{cut}) | [Z(\mathcal{U}_t^{cut}) = z_1] \not\equiv X(\mathcal{U}_t^{cut}) | [Z(\mathcal{U}_t^{cut}) = z_2],$$

where $z_1 = \zeta^{-1}_{Z(\mathcal{U}_t^{cut})}(\lambda_1)$ and $z_2 = \zeta^{-1}_{Z(\mathcal{U}_t^{cut})}(\lambda_2)$. Thus, this would imply that $X(\mathcal{U}_t^{cut})$ and $Z(\mathcal{U}_t^{cut})$ are not conditionally independent given $f(Z(\mathcal{U}_t^{cut}))$, giving a contraction.

Finally, Theorem 12 guarantees that a cut $\mathcal{U}_T^{cut}$ in an NT-DCEG $\mathbb{C}$ also defines a cut at time T in every CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C}), t = T, T+1, \ldots$ and by definition

Equations 7.11, 7.12 and 7.13 remain valid. In this case, a cut $\mathcal{U}_T^{cut}$ does not invalidate the conditional independence properties of standard cuts in CEGs (Smith and Anderson, 2008, p. 55). The result therefore follows. ■

Analogously to the BN framework, these constructions now enable us to identify conditional independence structures embedded within an NT-DCEG $\mathbb{C}$ that hold for all values of conditioning variables. Despite the probability mass function of each variable associated with a cut $\mathcal{U}_t^{cut}$ often being different over time t, $t = N-1, N, \ldots$, the collection of conditional independence statements that can be read from them is nevertheless equivalent. This happens because by Definition 46 each cut $\mathcal{U}_t^{cut}$, $t = N-1, N, \ldots$, corresponds to the same set of positions in an NT-DCEG $\mathbb{C}$. This assertion is also valid for every CEG $\mathbb{C}_t \in \mathfrak{F}(\mathbb{C})$, $t = N-1, N, \ldots$, since each time-slice t in $\mathbb{C}_t$ has the same stage structure as that of the subgraph $\mathbb{D}_H \subset \mathbb{C}$ (Theorem 12). The concept of cut is illustrated in the example below. Note that some care is needed in reading conditional independences associated with the time-slice $N-1$ because a path in the initial graph $\mathbb{D}_I$ may have probability zero: for an example about this case see Section 7.7.

Example 3 (Dynamic Radicalisation Process - cont.). Figure 7.7 depicts the 2T-DCEG associated with Example 3 and its corresponding stages. Take the cut $\mathcal{U}^{cut}$=$\{u_{13}, u_{14}\}$ for $t = 1, 2, \ldots$. The variable $X(\mathcal{U}_t^{cut})$ then corresponds to the initial variable Transfer. The variable $Q(\mathcal{U}_t^{cut})$ whose state space is given by $\mathbb{Q}(\mathcal{U}_t^{cut}) = \{1, 2\}$, such that $\zeta_{Q(\mathcal{U}_t^{cut})}(1) = u_{13}$ and $\zeta_{Q(\mathcal{U}_t^{cut})}(2) = u_{14}$, provides us an reinterpretation of the initial variable Radicalisation R.

In this case the variable $Q(\mathcal{U}^{cut})$ tell us that the variable R can be collected and analysed as a binary variable R^* that identifies whether a prisoner has adopted radicalisation ($Q = 2$) or not ($Q = 1$). We then have that

$$T(t+1) \perp\!\!\!\perp \boldsymbol{A} | (R^*(t+1), T(t) = n), \qquad t = 0, 1, \ldots, \tag{7.17}$$

where $\boldsymbol{A} = (N(t+1), \ldots, N(0), T(t-1), \ldots, T(0), R(t), \ldots, R(0))$.

Theorem 15 also guarantees that there is no information gain using the variable

Radicalisation with three categories to predict the probability of an inmate to be transfer to another prison once we have observed the variable $Q(\mathcal{U}^{cut})$.

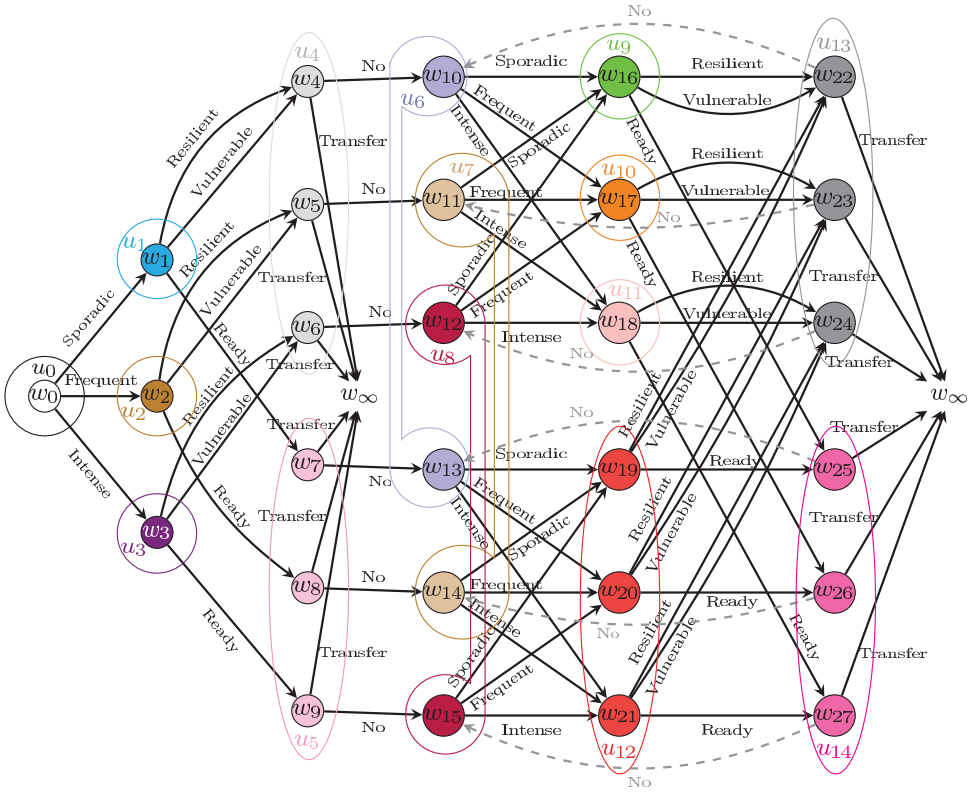


Figure 7.7: The 2T-DCEG associated with Example 3 and its corresponding stages. This 2T-DCEG is identical to that depicted in Figure 6.7. The stage structure is given by the following partition: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3\}$, $u_4 = \{w_4, w_5, w_6\}$, $u_5 = \{w_7, w_8, w_9\}$, $u_6 = \{w_{10}, w_{13}\}$, $u_7 = \{w_{11}, w_{14}\}$, $u_8 = \{w_{12}, w_{15}\}$, $u_9 = \{w_{16}\}$, $u_{10} = \{w_{17}\}$, $u_{11} = \{w_{18}\}$, $u_{12} = \{w_{19}, w_{20}, w_{21}\}$, $u_{13} = \{w_{22}, w_{23}, w_{24}\}$, $u_{14} = \{w_{25}, w_{26}, w_{27}\}$.

□

A cut allows us to describe conditional independences concerning developments 1-step ahead of situations in a staged tree. However, if a unit's developments over the next s time steps are of interest then we need to use an extended definition of fine cut to accommodate the time window s within which the present can affect the future. For this purpose, let $\mathcal{W}_t^{cut(s)}$ be the set of positions corresponding to $\mathcal{W}_t^{cut}$ when the focus is on s time steps ahead from the actual time. These new definitions are particular important because now a fine cut provides us with

a framework to identify global conditional independence structures that naturally arises from an NT-DCEG.

Note that when the time window s is greater than N–2 all necessary information to define a random variable associated with a fine cut can be obtained in a straight forward way from the NT-DCEG $\mathbb{C}$: $\mathcal{W}_t^{cut(s)} = \mathcal{W}_t^{cut}$. On the other hand, for shorter s we also need to use the CEG $\mathbb{C}_{g(t)+s} \in \mathfrak{F}(\mathbb{C})$ to define these variables, where g is a function such as $g(x) = x$, if $x \leq N-2$, and $g(x) = N-1$, otherwise. This is because the position set $\mathcal{W}_t^{cut(s)}$ associated with the current time t may be a coarser partition of situations than $\mathcal{W}_t^{cut}$ if s time-slices unfold from time t, when $s = 0, \ldots, N-2$; see the discussion in Section 7.3.

Let $\Lambda(\mathcal{W}_t^{cut}) = \cup_{w \in \mathcal{W}_t^{cut}} \Lambda(w, t)$, where $\Lambda(w, t)$ denotes the set of all walks

$$\lambda = (w_0, \ldots, w(t)) \subset \mathbb{C},$$

such that each walk λ passes through cycle temporal edges exactly $h(t)$ times. Also let $\Lambda_s(w)$ be the set of all walks that unfolds from w over s time-slices in $\mathbb{C}$ and $\xi(\lambda)$ be the sequence of events associated with a walk λ. Finally, let $\Xi(\mathcal{W}_t^{cut(s)}) = \cup_{w \in \mathcal{W}_t^{cut(s)}} \{\xi(\lambda); \lambda \in \Lambda_s(w)\}$ denote the set of sequences of events $\xi(\lambda)$ that can unfold from $\mathcal{W}_t^{cut}$ over s time-slices. When s is equal to zero, $\Xi(\mathcal{W}_t^{cut(s)})$ denotes the set of developments $\xi(\lambda)$ from $\mathcal{W}_t^{cut}$ during the current time-slice t. Analogously to a cut, I can now define three useful random variables that assume values over time t and time step s, $t = 0, 1, \ldots$ and $s = N-1, N, \ldots$, as follows:

1. $X(\mathcal{W}_t^{cut(s)})$ is the downstream random variable of $\mathcal{W}_t^{cut}$ in $\mathbb{C}$ whose state space is the set $\mathbb{X}(\mathcal{W}_t^{cut(s)}) = \{1, 2, \ldots, |\Xi(\mathcal{W}_t^{cut(s)})|\}$, such that there exists a bijection

   $$\varpi_{X(\mathcal{W}_t^{cut(s)})} : \mathbb{X}(\mathcal{W}_t^{cut(s)}) \to \Xi(\mathcal{W}_t^{cut(s)}).$$

   Its probability mass function $\pi_X(x)$ is defined by

   $$\pi_X(x) \propto \sum_{\bar{\lambda} \in \Lambda_x(\mathcal{W}_t^{cut(s)})} \prod_{\substack{w \in \bar{\lambda} \\ w \neq l(\bar{\lambda})}} \pi(w'(w)|w), \quad x \in \mathbb{X}(\mathcal{W}_t^{cut(s)}), \tag{7.18}$$

   where $\Lambda_x(\mathcal{W}_t^{cut(s)}) = \{\bar{\lambda} = (\lambda_*, \lambda) \subseteq \mathbb{C}; \lambda_* \in \Lambda(\mathcal{W}_t^{cut})$ and $\xi(\lambda) = \varpi_X(x)\}$ is

the set of all walks $\bar{\lambda}$ in $\mathbb{C}$ that have two disjoint sub-walks $\lambda_* \in \Lambda(\mathcal{W}_t^{cut})$ and λ, such that λ unfolds from λ_* over s time-slices and $\xi(\lambda) = \varpi_X(x)$.

2. $Q(\mathcal{W}_t^{cut(s)})$ is the separator random variable whose state space is given by the set $\mathbb{Q}(\mathcal{W}_t^{cut(s)}) = \{1, 2, \ldots, |\mathcal{W}_t^{cut}|\}$, such that there is a bijection

$$\varpi_{Q(\mathcal{W}_t^{cut})} : \mathbb{Q}(\mathcal{W}_t^{cut}) \to \mathcal{W}_t^{cut}.$$

Its probability mass function $\pi_Q(q)$ is proportional to the sum of all the monomials in primitives associated with $\Lambda(w, t), w \in \mathcal{W}_t^{cut}$. Symbolically then,

$$\pi_Q(q) \propto \sum_{\lambda \in \Lambda(\varpi_Q(q), t)} \prod_{\substack{w \in \lambda \\ w \neq l(\lambda)}} \pi(w'(w)|w), \quad q \in \mathbb{Q}(\mathcal{W}_t^{cut}). \tag{7.19}$$

3. $Z(\mathcal{W}_t^{cut(s)})$ is the upstream random variable of $\mathcal{W}_t^{cut}$ in $\mathbb{C}$ whose state space consists of the set $\mathbb{Z}(\mathcal{W}_t^{cut}) = \{1, 2, \ldots, |\Lambda(\mathcal{W}_t^{cut})|\}$, such that there is a bijection

$$\varpi_{Z(\mathcal{W}_t^{cut})} : \mathbb{Z}(\mathcal{W}_t^{cut}) \to \Lambda(\mathcal{W}_t^{cut}).$$

Its probability mass function is proportional to each monomial in the primitives corresponding to a walk that constitutes its state spaces. Explicitly,

$$\pi_Z(z) \propto \prod_{\substack{w \in \lambda \\ w \neq l(\lambda)}} \pi(w'(w)|w), \quad z \in \mathbb{Z}(\Lambda(\mathcal{W}_t^{cut})). \tag{7.20}$$

where $\lambda = \varpi_{Z(\mathcal{W}_t^{cut})}(z)$.

Observe that again '$=$' can substitute '$\propto$' in the equations above if the NT-DCEG does not have a sink position.

To define these three variables when $t = 0, 1, \ldots$ and $s = 0, \ldots, N-2$, take a partition $\beth_t^s = \{\beth_{t,1}^s, \ldots, \beth_{t,K}^s\}$ of $\mathcal{W}_t^{cut}$. Recall from Section 7.3 that the position structure of $\mathbb{C}_{g(t)+s}$ at time $g(t)$ results from an application of the vertex contraction operator Φ (Definition 42) and so naturally yields a partition $\beth_t^s = \{\beth_{t,1}^s, \ldots, \beth_{t,K}^s\}$ over $\mathcal{W}_t^{cut}$ according to the merged vertices. Now set $\mathcal{W}_t^{cut(s)} = \beth_t^s$. For all $t = -1, 0, \ldots$, the definitions of random variables $X(\mathcal{W}_t^{cut(s)})$ and $Z(\mathcal{W}_t^{cut(s)})$, $s = 0, \ldots, N-2$, are identical to the variable X and Z associ-

ated with $\mathcal{W}_t^{cut(s)}, s = N-1, N, \ldots$. It then follows that Equations 7.18 and 7.20 remain valid when $s = 0, \ldots, N-2$.

Of course, the variable $Q(\mathcal{W}_t^{cut(s)})$ has to be redefined appropriately since its state space is now given by $\mathbb{Q}(\mathcal{W}_t^{cut(s)}) = \{1, 2, \ldots, |\beth_t^s|\}$, , such that there is a bijection $\varpi_{Q(\mathcal{W}_t^{cut(s)})} : \mathbb{Q}(\mathcal{W}_t^{cut(s)}) \to \beth_t^s$. Its probability mass function $\pi_Q(q)$ corresponds to the sum of all probabilities mass functions defined by Equation 7.19 associated of a position in $\beth_{t,i}^s$. Symbolically we therefore have that

$$\pi_Q(q) \propto \sum_{w \in \varpi_Q(q)} \sum_{\lambda \in \Lambda(w,t)} \prod_{\substack{\bar{w} \in \lambda \\ \bar{w} \neq l(\lambda)}} \pi(w'(\bar{w})|\bar{w}), \quad q \in \mathbb{Q}(\mathcal{W}_t^{cut(s)}). \tag{7.21}$$

These random variables enable us to read a large collection of conditional independence statements between vectors of functions of primitive random variables embedded into the NT-DCEG topology. This is because a fine cut is based on positions that gather situations in a staged tree all of whose future developments are equivalent. Theorem 16 tells us that a unit's future unfoldings are independent from the whole set of its past events given that the available information on it constitutes a fine cut. It also guarantees that, given a fine cut at time t and time-horizon s, a function of upstream variables that makes all the corresponding downstream variables conditionally independent from upstream variables must constitute a fine cut.

Theorem 16. *Take a fine cut $\mathcal{W}_t^{cut}$, $t = -1, 0, 1, \ldots$, in an NT-DCEG $\mathbb{C}$. For every $s = 0, 1, \ldots$, we have that*

$$\boldsymbol{X}(\mathcal{W}_t^{cut(s)}) \perp\!\!\!\perp \boldsymbol{Z}(\mathcal{W}_t^{cut(s)})|\boldsymbol{Q}(\mathcal{W}_t^{cut(s)}). \tag{7.22}$$

Additionally, if a function $f(Z(\mathcal{W}_t^{cut(s)}))$ satisfies

$$X(\mathcal{W}_t^{cut(s)}) \perp\!\!\!\perp Z(\mathcal{W}_t^{cut(s)})|f(Z(\mathcal{W}_t^{cut(s)})), \tag{7.23}$$

then $Q(\mathcal{W}_t^{cut(s)})$ is a function of $f(Z(\mathcal{W}_t^{cut(s)}))$ with probability one. These results also hold when a fine cut $\mathcal{W}_T^{cut(s)}$ is defined in a CEG $\mathbb{C}_{t+s} \in \mathfrak{F}(\mathbb{C})$, $t = T, T+1, \ldots$.

Proof. We can assert immediately from the construction that given a value q for $Q(\mathcal{W}_t^{cut(s)})$, then any random variable associated with $\varpi_{Q(\mathcal{W}_t^{cut(s)})}(q)$ is completely defined. So none of the w_0-to-$\varpi_{Q(\mathcal{W}_t^{cut(s)})}(q)$ walks can bring any additional information on the realization of the random variable $X(\mathcal{W}_t^{cut(s)})$. Thus,

$$X(\mathcal{W}_t^{cut(s)}) \perp\!\!\!\perp Q(\mathcal{W}_t^{cut(s)})|Z(\mathcal{W}_t^{cut(s)}).$$

Now suppose that $Q(\mathcal{W}_t^{cut(s)})$ is not a function of $f(Z(\mathcal{W}_t^{cut}))$ with probability one. Then, for $s = N-1, N, \ldots$, there are at least two non-zero probability walks λ_1 and λ_2 in $\Lambda(\mathcal{W}_t^{cut})$ such that $l(\lambda_1) \neq l(\lambda_2)$, $f(z_1) = f(z_2)$ and

$$X(\mathcal{W}_t^{cut(s)})|[Z(\mathcal{W}_t^{cut(s)}) = z_1] \not\equiv X(\mathcal{W}_t^{cut(s)})|[Z(\mathcal{W}_t^{cut(s)}) = z_2],$$

where $z_1 = \varpi^{-1}_{Z(\mathcal{W}_t^{cut})}(\lambda_1)$ and $z_2 = \zeta^{-1}_{Z(\mathcal{W}_t^{cut})}(\lambda_2)$. Thus, this would imply a contraction because $X(\mathcal{W}_t^{cut(s)})$ and $Z(\mathcal{W}_t^{cut(s)})$ were not conditionally independent given $f(Z(\mathcal{U}_t^{cut}))$. For time-horizon s, $s = 0, \ldots, N-2$, the proof is completely analogous to that one except that the condition $l(\lambda_1) \neq l(\lambda_2)$ needs to be rewritten as follows: $l(\lambda_1)$ and $l(\lambda_2)$ are in different set of the partition Ξ_t^s. The result then follows.

From Theorem 12 we can assert that in an NT-DCEG $\mathbb{C}$ a fine cut $\mathcal{W}_t^{cut(s)}$ also defines a fine cut $\mathcal{W}_t^{cut(s)}$ at time T in every CEG $\mathbb{C}_{t+s} \subset \mathfrak{F}(\mathbb{C})$, $t = T, T+1, \ldots$, $s = 0, 1, \ldots$. By construction, Equations 7.18, 7.19, 7.21 and 7.20 also hold. The result then follows due to the conditional independence properties of standard fine cuts in CEGs (Smith and Anderson, 2008, p. 61). ■

Thus, the units' behaviours may present important differences in the medium and long time ($s \geq N-1$) but may be undistinguishable in the short term ($s \leq N-2$). For analogous reasons to those discussed for a cut, a fine cut $\mathcal{W}_t^{cut(s)}$ entails the same set of conditional probability statements when $t = N-1, N, \ldots$ for a given time-horizon s and when $s = N-1, N, \ldots$ for a given time-slice t. Thus, for any 2T-DCEG $\mathbb{C}$ all these constructions associated with a fine cut only require the three CEGs $\mathbb{C}_i, i = 0, \ldots, 2$.

Recall that Theorem 15 tells us that a cut gathers all the necessary information to predict the immediately development of a unit in a process. Theorem 16 guarantees

that the future events are independent from past events given the actual state of a process. Therefore, these results enable us to use cuts and fine cuts to deduce some conditional independence statements given some observed effects in a way that extends the BN and DBN framework using the d-separation theorem. This happens because a Laminated DCEG can enrich the conditional independence hypotheses depicted in its corresponding DBN with context-specific deductions and the time horizon s. These links are further discussed through the example below using the concept of fine cut.

Example 3 (Dynamic Radicalisation Process - cont.). Return to the 2T-DCEG $\mathbb{C}$ depicted in Figure 6.7. Assume that domain experts are interested in exploring the impacts of social connections on the risk of inmate's radicalisation and transfer at the current time-slice $t, t = 1, 2, \ldots$, given that the past information is completely available. For this propose, we may take the fine cut

$$\mathcal{W}_t^{cut} = \{w_{10}, w_{11}, w_{12}, w_{19}, w_{20}, w_{21}\}.$$

Since the experts' focus is on the current time-slice, we also need to set the time window s equal to 0. As discussed previously, this time horizon then requires us to replace the fine cut $\mathcal{W}_t^{cut}$ by $\mathcal{W}_t^{cut(0)} = \beth_t^0$. Using the CEG $\mathbb{C}_2 \in \mathfrak{F}(\mathbb{C})$ showed in Figure 7.5, we can see that $\beth_t^0 = \{\beth_{t,i}^0, i = 1, \ldots, 4\}$, where $\beth_{t,1}^0 = \{w_{10}\}$, $\beth_{t,2}^0 = \{w_{11}\}$, $\beth_{t,2}^0 = \{w_{12}\}$ and $\beth_{t,4}^0 = \{w_{19}, w_{20}, w_{21}\}$.

The random variable $Q(\mathcal{W}_t^{cut(0)})$ then has four states $\{1, \ldots, 4\}$, such that $\varpi_{Q(\mathcal{W}_t^{cut(0)})}(i) = \beth_{t,i}^0$, $i = 1, \ldots, 4$. These have the following interpretations: categories $1, 2$ and 3 characterise non-radicalised prisoners whose social networks were classified, respectively, as Sporadic, Frequent and Intense at time $t-1$; and category 4 represents a prisoner adopting radicalisation at time $t-1$. The variable $X(\mathcal{W}_t^{cut(0)})$ has 24 states associated bijectively with the set of sequences of events

$$\Xi(\mathcal{W}_t^{cut(0)}) = \{(R, T), (N, R, T); N = s, f, i, R = r, v, a \text{ and } T = n, t\}.$$

So given a fine $\mathcal{W}_t^{cut(0)}$ we have 24 possible outcomes in the end of this time-slice.

Additionally this fine cut $\mathcal{W}_t^{cut(0)}$ tells us that the social network of an adopting

prisoner at time t affects neither his radicalisation process nor his transfer probability. However this is not true if the inmate does not adopt the radicalisation. Note that this kind of context-specific d-separation statement cannot be directly deducted from a BN in Figure 2.2 or its corresponding undirected moralised graph. For a larger time length, $s = 1, 2, \ldots$, we can see directly from the 2T-DCEG (Figure 6.7) that the future developments of a prisoner depend on the random vector $\boldsymbol{Q}(\mathcal{W}_t^{cut(s)})$ whose set of states are given by $\mathcal{W}_t^{cut}$. So now the current status of an adopting prisoner's network has an impact in his unfolding events.

Note that analogous conclusions could have been obtained if we had used the fine cut $\mathcal{W}_t^{cut} = \{w_{16}, \ldots, w_{21}\}$. However, in this case the interpretation of the variable $Q(\mathcal{W}_t^{cut(0)})$ would be based on the actual (time t) social network of an inmate instead of his previous social classification at time $t-1$. This would enable us to easily update our judgements at time t as new information about the social contacts of an inmate is collected. □

7.7 Interrogating a simple 2T-DCEG model

Here I will present and analyse a simple escalation 2T-DCEG $\mathbb{C}$ (Figure 7.8), which models a radicalisation process of inmates using only two variables: Network (N) and Radicalisation (R). In this example the categories *sporadic* and *frequent* corresponding to the variable N in Example 3 are merged into a single category *moderate* (m). To facilitate the interpretability in the subgraph $\mathbb{D}_H \subset \mathbb{C}$ the edges with probability zero are omitted whilst in the subgraph $\mathbb{D}_I \subset \mathbb{C}$ they are showed as dotted edges. Note that the zero-probability edges cannot be eliminated from $\mathbb{D}_I$ because they do not only support the probability map but also play a role as a legend through which we can read the unfolding process at time-slices t, $t = 1, 2, \ldots$ (Section 7.3).

The 2T-DCEG (Figure 7.8) is supported by an event tree without terminating events since there isn't a position w_∞. This means that all prisoners remain in the prison system over time. This also implies that the process has a unique stationary

distribution (Corollary 8). We can read directly from $\mathbb{C}$ that the variables N and R are not locally or instantaneously independent and so neither are stochastically independent of one another.

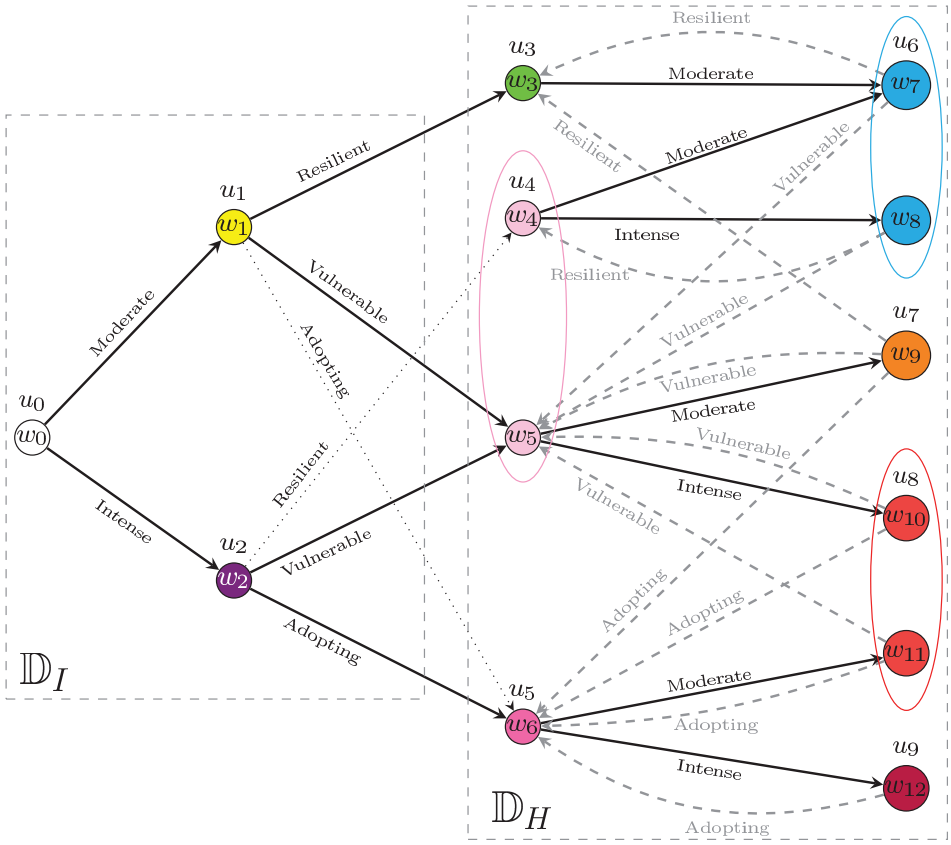


Figure 7.8: A possible 2T-DCEG for the radicalisation process of prisoners. In the graph $\mathbb{D}_I$ dotted edges are asssociated with probability zero. In the graph $\mathbb{D}_H$ these edges are omitted. The temporal edges are dashed and grey. Hotter colours implies higher risk of radicalisation. The stage structure is given by the following partition: $u_0 = \{w_0\}$, $u_1 = \{w_1\}$, $u_2 = \{w_2\}$, $u_3 = \{w_3\}$, $u_4 = \{w_4, w_5\}$, $u_5 = \{w_6\}$, $u_6 = \{w_7, w_8\}$, $u_7 = \{w_9\}$, $u_8 = \{w_{10}, w_{11}\}$, $u_9 = \{w_{12}\}$.

Note that a non-radicalised prisoner who has intense social contact with potential recruiters (stage u_4) at time $t, t = 0, 1, \ldots,$ has the same chance of keeping this socialisation pattern at time $t + 1$ regardless of being resilient or vulnerable to radicalisation at the previous time. Focusing on stage u_6, we see that the probability of a resilient prisoner to adopt radicalisation does not depend on his social network. Taking the stage u_8 we conclude that a vulnerable prisoner and

an adopting one at time $t, t = 0, 1, \ldots,$ whom maintain, respectively, intense and moderate social contacts with potential recruiters have the same deradicalisation probabilities (stage u_8).

To explore the radicalisation process over time take a cut $\mathcal{U}_t^{cut} = \{u_6, u_7, u_8, u_9\}$, $t = 1, \ldots,$ made up of the stages associated with variable R. Define the corresponding variables $X(\mathcal{U}_t^{cut})$, $Q(\mathcal{U}_t^{cut})$ and $Z(\mathcal{U}_t^{cut})$. The variable $X(\mathcal{U}_t^{cut})$ is then analogous to the initial variable N. The state space of variable $Q(\mathcal{U}_t^{cut})$ is given by $\{1, 2, 3, 4\}$, such that:

1. $\zeta_{Q(\mathcal{U}_t^{cut})}(1) = u_6$ - prisoners who were resilient to radicalisation at time $t-1$;
2. $\zeta_{Q(\mathcal{U}_t^{cut})}(2) = u_7$ - prisoners who were vulnerable to radicalisation at time $t-1$ and have kept moderate social contacts with potential radical recruiters at the current time t;
3. $\zeta_{Q(\mathcal{U}_t^{cut})}(3) = u_8$ - prisoners who were vulnerable to radicalisation and have had intense social contacts with potential recruiters at time t, and prisoners who adopted radicalisation at time $t-1$ but have kept moderate social contacts with potential recruiters at time t; and
4. $\zeta_{Q(\mathcal{U}_t^{cut})}(4) = u_9$ - prisoners who adopted radicalisation at time $t-1$ and have maintained intense social contacts with potential recruiters at the ongoing time-slice.

This implies that variable $Q(\mathcal{U}_t^{cut})$ partitions the prison population into four groups according to their risk of adopting radicalisation. Theorem 15 guarantees that this the maximum information obtained from $\mathbb{C}$: the current risk of an inmate to adopt radicalisation is completely defined by these four groups of prisoners given that the values of variable R at the previous time-slice and variable N at the ongoing time-slice are known. In this case the information about social contacts in the previous time-slice does not bring any further gain to our analysis.

Of course, this result can be refined using the concept of fine cut. Now assume that we are interested in exploring how the radicalisation process develops over one time-slice ahead. For this purpose, take the fine cut $\mathcal{W}_t^{cut} = \{w_7, \ldots, w_{12}\}$,

$t = 1, 2, \ldots$, and set the time length at $s = 1$. Next define the corresponding variables $X(\mathcal{W}_t^{cut(1)})$, $Q(\mathcal{W}_t^{cut(1)})$ and $Z(\mathcal{W}_t^{cut(1)})$. This is facilitated if we construct, for instance, the CEG model $\mathbb{C}_2 \subset \mathfrak{F}(\mathbb{C})$ (Figure 7.9).

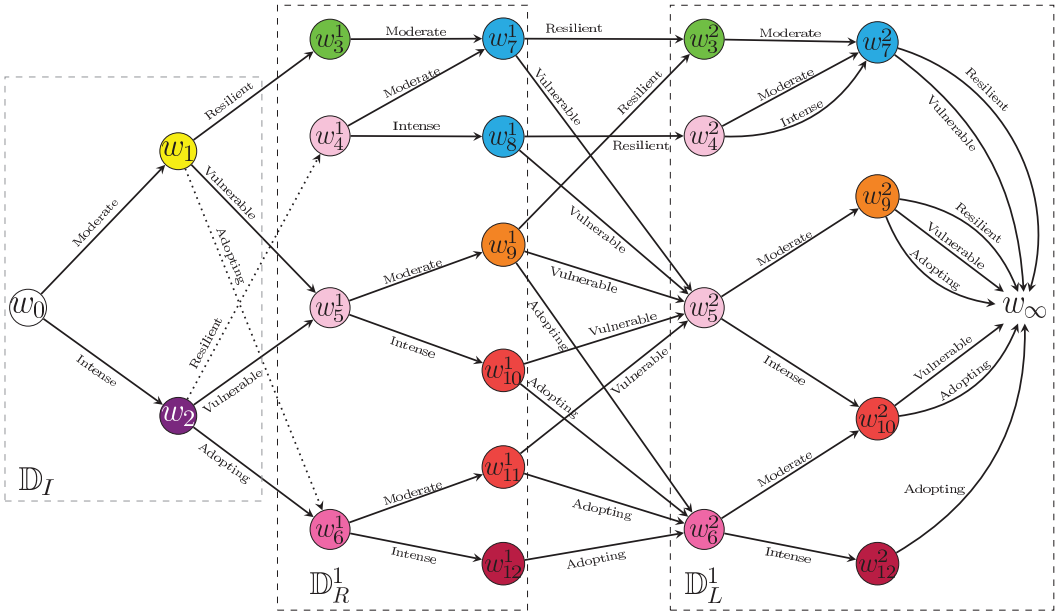


Figure 7.9: The CEG $\mathbb{C}_2$ associated with the 2T-DCEG depicted in Figure 7.8.

The random variable $Q(\mathcal{W}_t^{cut(1)})$ has seven states $\{1, \ldots, 7\}$, such that

$$\varpi_{Q(\mathcal{W}_t^{cut(1)})}(i) = w_{6+i}, i = 1, \ldots, 7.$$

The variable $X(\mathcal{W}_t^{cut(1)})$ has 46 states associated bijectively with the set of possible sequences of events that can unfold from the set of positions $\{w_7^1, \ldots, w_{12}^1\}$ in CEG $\mathbb{C}_2$. Theorem 16 enables us to read some useful context-specific conditional independences that might be kept hidden at first glance. For example, if an inmate is classified as having the highest risk of adopting radicalisation (position w_{12}) at current time, then he will not be resilient in one time-slice ahead and his social networks will play a key role for his deradicalisation. We can also see that all other type of inmates may be radicalised in one time-slice ahead driven by their social networks. These interpretations may then prompt decision makers to adopt policy that encourages the segregation of potential recruiters from the rest of the prison population.

Now assume the risk of radicalisation defined by the variable $Q(\mathcal{U}_t^{cut})$ above. Note

that a prisoner with the lowest risk of adopting radicalisation (positions w_7 or w_8) will not be at the highest risk of adopting radicalisation in one time-slice. His radicalisation process is necessarily gradual over time since he will need to pass through the intermediate risk positions w_9, w_{10} or w_{11}. Moreover, if he is resilient to radicalisation at the end of time-slice t, then he will not adopt radicalisation at the end of the next time with probability one. The deradicalisation process also follows an escalated dynamic where a prisoner at high risk of adopting radicalisation (positions w_{10}, w_{11} and w_{12}) does not deradicalised (position w_7 and w_8) without passing through the intermediate risk position w_9. Furthermore if a prisoner is identified as adopting radicalisation at the end of time-slice t, then he will not be resilient to radicalisation at the end of the next time-slice regardless of his social networks. So we can see that a prisoner cannot be radicalised or deradicalised over two risk categories during only one time-slice.

Chapter 8

Discussion

In this thesis, I have proposed a more computationally efficient algorithm for propagating new information over a CEG model based on a earlier algorithm developed by Thwaites and Smith (2006a). I have also formally extended the characterisation of Dirichlet prior distributions from a model space spanned by only one event tree (Freeman and Smith, 2011a) to a model space yielded by a collection of event trees. In close analogy to BN model selection (Heckerman and Geiger, 1995, Geiger and Heckerman, 1997), this was directly done by assuming the conditions of structural possibility and likelihood equivalence over the set of the corresponding staged trees.

I have also showed through an original marketing analysis associated with a process of booking a tourist train how a CEG model can be useful to analyse real-world problems that are characterised by highly asymmetric developments and some context-specific conditional statements. As argued in Section 2.6 these structures cannot be easily accommodated in BN models and their variants.

However providing us with a very flexible and representative framework the space of CEG models becomes massive even in problems with a very moderate number of situations in the event tree. Some strategies to circumvent this issue were presented in Section 4.4 using some approximative model search algorithms and parallel computing. In fact the explosive number of CEG models usually forces

us to adopt a heuristic strategy to perform CEG model selection as previously recognised by other authors (Silander and Leong, 2013, Cowell and Smith, 2014).

Building on a previous greedy model search algorithm proposed by Freeman and Smith (2011a) and using the new concept of hyper-stages I have recently developed a more efficient algorithm to search over the CEG model class. My algorithm allows me to reduce the computational complexity from quartic order in terms of size of the CEG model space to a quadratic order of it. I have also shown that the hyper-staged structures enable us to embed some domain information within the model search algorithm. This often reduces the size of CEG model spaces and tends to provide results that are more appealing for domain experts and decision makers. These results will be reported in a later paper.

In Chapter 5 I investigated some undesirable phenomena that may happen when standard CEG model selection is conducted using an agglomerative model search algorithm such as the AHC or OAHC algorithms. To address these problems I proposed three new classes of NLPs for CEGs (fp-NLPs, pp-NLPs, pm-NLPs) and discussed some of their properties associated with greedy model search strategies.

A formal framework to adapt the pm-NLPs to the OAHC algorithm was also developed. I argued that pm-NLPs are the best option since they are more prone to find reliable and parsimonious CEGs than Dirichlet local priors or products NLPs. They also appear to explore more wisely the asymmetric context-specific conditional independences that may be present in the data. I also showed that both families of product NLPs can bias inappropriately the OAHC algorithm and demand greater computational cost as regards implementation. This may happen because the product NLPs accommodate the separation measure between models based on a larger collection of stages than one used to define the search neighbourhood (a pair of stages) of the OAHC algorithm.

However the product NLPs might still be computationally practicable and statistically effective if combined with other heuristic search algorithms that enlarge the search neighbourhood. For example, the NLP frame for CEGs can stimu-

late some future developments using the weighted MAX-SAT algorithm (Cussens, 2008, Liverani et al., 2010). Being an integer programming formulation, the search locality of this algorithm can be defined in terms of two or more stages whilst the computational cost is kept under reasonable control. These enlargements in the neighbourhood provide background to envisage more efficient search strategies as well as to explore some peculiarities of real-world problems.

In contexts where the search is not conducted in a way that is either sequential or pairwise, pp-NLPs are more appropriate than pm-NLPs and can be used as a good approximation of fp-NLPs. For instance, I recently modelled the CHDS data set using an integer programming formulation based on Dirichlet local priors. Using my own solver based on Lagrangian relaxation (Wolsey, 1998, Wolsey and Nemhauser, 2014), I was able to search the CHDS model space very efficiently. I intend to extend this formulation to accommodate pp-NLPs and results will be reported in the future.

Another extension that has not been pursued in this thesis is to use the dynamic programming framework in conjunction with the full product NLPs and some heuristic strategies such as those discussed in Section 4.4. Despite it being possible to write down analytically the fp-NLPs in closed form, its computational implementation is much more complex and demands more computational resources in terms of memory, processing time and coding. These challenges are even more pronounced in discrete high-dimensional applications. Whilst in Section 5.3 I developed the theoretical background for this objective, it is now vital to design bespoken algorithms that optimise the computational costs of time and memory. This will enable us to scale up the applications to medium size instances, and to keep under control the modelling and coding complexities.

A dynamic programming algorithm adapted for fp-NLPs would provide useful advice to assess the results given by the OAHC algorithm when used in conjunction with the pm-NLPs (Section 5.3.1). It could also constitute a reference to be used to analyse the robustness of the OAHC algorithm to the values of the hyper-

parameter $\bar{\alpha}$. Although the empirical experiments demonstrated that pm-NLPs are a promising method to reduce common difficulties associated with setting this hyper-parameter, I recognise that further studies are needed to address these issues.

Another promising and unexplored research stream is to explore the CEG model space using stochastic algorithms. Observe that these algorithms often use local moves which will tend to face analogous drawbacks to those described for the greedy search algorithms using standard Dirchlet priors. It is then to be expected that in these cases pm-NLPs and pp-NLPs will avoid these incoveniences for similar reasons to those presented for the OAHC algorithm. Among its advantages, these priors also speed up the learning rate, disposing of unlikely models at a quicker pace than standard CEG model selections. These advances will be important if we wish to work with more flexible CEG classes in which stages at different levels may be merged. Whilst this relaxation allows us to explore highly asymmetric CEGs, it also increases our search space sharply, making stochastic strategies indeed a good option.

The Bayesian CEG model selection based on pm-NLPs is particularly suitable to analyse domains where the client is interested in gaining insights of how factors or variables affect an elicited unfolding process. It is also helpful to explore causal relations if it is combined with a BN search in a way that advances the strategy first proposed by Barclay et al. (2013) and discussed here in Section 4.2.2. As posed by Korb and Nicholson (2011), causal problems involved two questions: finding the best variable order and searching for the best graphical structure given a variable order. In this case, the first problem is solved in the BN context whilst the OAHC algorithm using pm-NLPs is a good alternative to the second challenge since it allows us to look for asymmetric context-specific statements in a faster and more reliable way.

A further generalisation of this model search framework that combines a heuristic strategy and NLPs with more general model classes also appears encouraging. For

example, it is not difficult to generalise these three families of NLPs for discrete BNs. I have customised the dynamic programming algorithm for BN model selection (Silander and Myllymaki, 2006) using the full product NLPs. Being a smaller model space it is less challenging to elicit the fp-NLPs analytically and implement them compared to the CEG framework. For the sake of brevity the computational results associated with the CHDS data set using fp-NLPS for BN model selection are not presented in this thesis despite looking very promising.

In Chapters 6 and 7, I argued that an NT-DCEG is able to encode many asymmetric and context-specific independence structures that characterise process observed over discrete time. I have shown that these can be read directly using the algorithmic tools I developed in Section 7.4. In analogy to the interrogation methods used for a BN, the deductions from a DCEG model can be fed back to domain experts for verification or criticism. This process will continue until the hypothesised model is requisite (Smith, 2010, Phillips, 1984, 2007); i.e until no obvious inadequacies in the implications of the model could be found. In this way the plausibility of the qualitative implications of a hypothesised model can be examined *before* any costly quantitative population of the graphical probability model takes place.

I have shown that this new dynamic class of models is compact. It is therefore sufficient to provide us with a framework for fast propagation of evidence and for model selection. Adopting a Bayesian approach (Barclay et al., 2015), for example, we can directly extend the propagation algorithm (Thwaites et al., 2008) and model selection methods (Freeman and Smith, 2011a, Barclay et al., 2013, Cowell and Smith, 2014) that exist for CEGs.

However, it is immediately apparent that the NT-DCEG model spaces has an order of magnitude greater than those discussed for CEGs. Therefore it is vital to design efficient algorithms that make good use of computational time and memory to search these model spaces. In particular the OAHC algorithm using non-local priors looks very promising for NT-DCEG model spaces. Such algorithms have already been used to search a dynamic version of the CHDS data set and the

results will be reported in future work.

It was demonstrated in Section 7.4 that the conditional independences in an NT-DCEG can also be interpreted in terms of Granger noncausality. However these relationships need to be developed further. For example, by identifying a fine cut $\mathcal{W}^{cut(s)}$ we are able to expose the conditional independences across different time steps s. This then leads us to a direct link with Granger noncausality as it applies to different time horizons (Dufour and Renault, 1998) and gives us a new graphical framework which appears to be able to distinguish between short-run and long-run causal mechanisms.

Here the Granger noncausality is defined with respect to the whole set of past events $\mathcal{E}^{(t)}$. However, we may want to focus our attention on a particular subset of events $\mathcal{E}_*^{(t)} \subset \mathcal{E}^{(t)}$. Similarly, the conditions determining the Granger noncausal relations with respect to a proper subset of events $\mathcal{E}_*^{(t)}$ in a DCEG still remain unexplored as do the types of assumptions about local independences that are needed in order to assess the causal effect arising from an intervention on the system. As discussed for path diagrams applied to time series (Eichler and Didelez, 2010), these developments demand the combination of the Granger noncausality idea (Granger, 1969) and Pearl's stronger causality concept (Pearl, 2009) presented in Thwaites et al. (2010) and Thwaites (2013). An NT-DCEG also appears to provide a useful framework for deriving contemporaneous causal relations (Granger, 1988, Ltkepohl, 1993).

A larger family of NT-DCEG models can also be obtained if a position on a DCEG implies only the Markov condition between situations in different time-slices but not the time-homogeneity. In this case, we can propose a learning framework that connects time-slices and also provides flexibility to handle local-time disturbances in a process whilst the graphical representation remains identical to that of a standard NT-DCEG presented above.

In the future I plan to extend my object-recursive approach and the *tree* objects developed in Section 6.1 to define a continuous time DCEG (CT-DCEG). This

new family of DCEGs will enable us to systematically construct models primarily designed to describe how and when events might happen during irregular time-step transitions. Such CT-DCEG models should then extend the continuous time BN (Nodelman et al., 2002, 2003) and further explore the link between a general DCEG with holding times and semi-Markov processes (Barclay et al., 2015).

I will demonstrate in a later paper that the process-driven objects defined here can also be used to define an Object-Oriented CEG/DCEG and so provide a generalisation of Object-Oriented BNs (Koller and Pfeffer, 1997, Bangsø and Wuillemin, 2000). I believe that this development will facilitate the knowledge engineering process using event trees and the reusability of computational codes, particularly in large and complex real-world applications.

Finally, I have noted that criminal dynamics and radicalisation in prisons are challenging processes to model since their developments tend to be highly asymmetric yielding event trees with many zero-count partitions. It is becoming clear that CEGs using NLPs are able to circumvent these issues and to provide reliable support to decision-makers. I intend to customise the current CEG tools for these kinds of more complex and realistic problems, which include the prevention of crimes and early intervention to stop the radicalisation process. I also believe that a dynamic approach will be a natural development for my models in the criminal and radicalisation domains.

Appendix A

List of the CEG/DCEG notation

Here I present a list of the most important CEG and DCEG notation for the purpose of this thesis. In parenthesis I point out the chapter where a symbol was first introduced in this thesis.

$a(\boldsymbol{y})$	$a(\boldsymbol{y}) = \log\Gamma(\sum_{i=1}^{n} y_i)$, where $\boldsymbol{y} = (y_1, \ldots, y_n)$ (Chapter 3)
$a(s_i, t)$	antecedent situation of a situation s_i such that $a(s_i, t) \in \mathcal{S}(t)$ (Chapter 6)
$a(V)$	set of all vertices in a graph that are antecedents of at least one vertex in V (Chapter 7)
$b(\boldsymbol{y})$	$b(\boldsymbol{y}) = \sum_{i=1}^{n} \log\Gamma(y_i)$, where $\boldsymbol{y} = (y_1, \ldots, y_n)$ (Chapter 3)
e_{ij}	outgoing edge j from a situation s_i in an event tree (Chapter 3)
$e_i(w)$	outgoing edge i from a position w in a CEG/DCEG (Chapter 3)
l	leaf vertex in an event tree (Chapter 3)
$l(\mathcal{T})$	set of leaf nodes of an event tree $\mathcal{T}$ (Chapter 6)
m_{ij}	transition probability from a state x_i to a state x_j in a Markov chain (Chapter 7)
$q(\mathbb{C})$	prior probability of a CEG $\mathbb{C}$ (Chapter 4)
$q_{LP}(\boldsymbol{\pi}_i)$	Dirichlet local prior probability distribution associated with a stage u_i (Chapter 5)
$q_{NLP}(\boldsymbol{\pi}_i)$	non-local prior probability distribution associated with a stage u_i (Chapter 5)

s	situation in an event tree (Chapter 3)
$s(t)$	situation that happens in time-slice t (Chapter 6)
$s(\boldsymbol{z}^{(k)})$	situation in a $\boldsymbol{\mathcal{Z}}-$compatible event tree $\mathcal{T}(\boldsymbol{Z}(I))$ corresponding to an element $\boldsymbol{z}^{(k)} = (z_{i_1}, z_{i_2}, \ldots, z_{i_k})$ in $\mathbb{Z}^{(k)}(I)$ (Chapter 4)
u	stage in a CEG/DCEG model (Chapter 3)
w	position in a CEG/DCEG model (Chapter 3)
$w_i(t)$	position of a DCEG/CEG associated with a time-slice t (Chapter 7)
w_∞	sink position in a CEG/DCEG (Chapter 3)
$\boldsymbol{x}_i$	sample vector corresponding to a stage u_i (Chapter 3)
$\bar{x}_i$	total number of units that visit stage u_i (Chapter 3)
$\boldsymbol{x}^{(n)}$	sample of size n (Chapter 5)
x_{ij}	number of units that transverse the stage u_i using its outgoing j (Chapter 3)
$\boldsymbol{z}^{(k)}$	element $\boldsymbol{z}^{(k)} = (z_{i_1}, z_{i_2}, \ldots, z_{i_k})$ in $\mathbb{Z}^{(k)}(I)$ (Chapter 4)
$\boldsymbol{\mathcal{B}}$	block order of a set of random variables $\boldsymbol{\mathcal{Z}}$ (Chapter 4)
$\boldsymbol{\mathcal{B}}_i$	i^{th} block of $\boldsymbol{\mathcal{B}}$ (Chapter 4)
B_n	n^{th} Bell number (Chapter 4)
$B(\boldsymbol{\alpha})$	normalization constant for the Dirichlet distribution parametrised by the vector $\boldsymbol{\alpha}$ (Chapter 5)
$\boldsymbol{\mathcal{C}}$	CEG model space (Chapter 4)
$\mathbb{C}$	CEG/DCEG model (Chapter 3)
$\mathbb{C}_t$	CEG spanned by the staged tree $\mathcal{ST}_t \subset \mathcal{ST}_\infty$ (Chapter 7)
$\mathbb{C}_u$	the transporter model of a CEG $\mathbb{C}$ (Chapter 3)
$\mathbb{C}_0$	CEG that has the finest stage structure supported by its event tree (Chapter 4)
$\mathbb{C}^+$	m-nested CEG in a CEG $\mathbb{C}$ (Chapter 4)
$\boldsymbol{\mathcal{C}}(\boldsymbol{Z}(I))$	set of SCEGs specified over $\mathcal{T}(\boldsymbol{Z}(I))$ (Chapter 4)
$\mathbb{D}^t_{l(a)}$	$\mathbb{D}^t_{l(a)} = \mathbb{D}^t_R \oplus \mathbb{G}_{2(t)}, t = N-1, \ldots, 2N-3$ (Chapter 7)
$\mathbb{D}^t_{l(b)}$	$\mathbb{D}^t_{l(b)} = \mathbb{D}^{t-N+2,t}_R \oplus \mathbb{G}_{2(t)}$, $t = 2N-2, 2N-1, \ldots$ (Chapter 7)

$\mathbb{D}_H$	cyclic subgraph of an NT-DCEG that represents the time-homogeneous developments of the process and then contains the cyclical temporal edges from time-slice t to $t+1, t = N–1, N, \ldots$ (Chapter 7)
$\mathbb{D}_I$	subgraph of an NT-DCEG that initialises the modelled process over the first $N–1$ time-slices (Chapter 7)
$\mathbb{D}_L^t$	$\mathbb{D}_L^t = \Phi(\mathbb{G}_{2(t-N+1)} \oplus \mathbb{D}_{L(b)}^t)$ (Chapter 7)
$\mathbb{D}_R$	graph obtained from $\mathbb{D}_H$ when its cyclical temporal edges are removed (Chapter 7)
$\mathbb{D}_{R(t)}$	graph obtained from $\mathbb{D}_R$ by a label transformation of its vertices (Chapter 7)
$\mathbb{D}_R^t$	$\mathbb{D}_R^t = \mathbb{D}_R^{N–1,t}$ (Chapter 7)
$\mathbb{D}_R^{N–1,N–1}$	$\mathbb{D}_R^{N–1,N–1} = \mathbb{D}_{R(N–1)}$ (Chapter 7)
$\mathbb{D}_R^{t_a,t_b}$	$\mathbb{D}_R^{t_a,t_b} = \mathbb{D}_{R(t_a)} \oplus \mathbb{G}_{2(t_a)} \oplus \mathbb{D}_{R(t_a+1)} \oplus \mathbb{G}_{2(t_a+1)} \oplus \ldots \oplus \mathbb{D}_{R(t_b-1)} \oplus \mathbb{G}_{2(t_b-1)} \oplus \mathbb{D}_{R(t_b)}$, $t_a < t_b$ (Chapter 7)
$Dir(\boldsymbol{\alpha})$	Dirichlet probability distribution with hyper-parameter $\boldsymbol{\alpha}$ (Chapter 3)
$\mathcal{E}$	particular set of past events (Chapter 7)
$\mathcal{E}^{(t)}$	collection of all sequences of events that happened up to the end of time-slice t (Chapter 7)
$\mathcal{E}_{(-\boldsymbol{X})}^{(t)}$	collection of all sequences of events that happened up to the end of time-slice t that excludes information with respect a random vector $\boldsymbol{X}$ (Chapter 7)
$E_{\dagger}$	set of cyclical temporal edges in a DCEG (Chapter 7)
$E(\Delta)$	set of direct edges associated with a tree object (Chapter 6)
$E_{\boldsymbol{\pi}}[f(\boldsymbol{\pi})]$	expectation of $f(\boldsymbol{\pi})$ that is calculated using the Dirichlet local prior of $\boldsymbol{\pi}$ (Chapter 5)
$E_{\boldsymbol{\pi}^*}[f(\boldsymbol{\pi})]$	expectation of $f(\boldsymbol{\pi})$ that is calculated using the posterior of $\boldsymbol{\pi}$ corresponding to a Dirichlet local prior (Chapter 5)

$\mathcal{F}(s)$	floret associated with a situation s in an event tree (Chapter 3)
$\mathfrak{F}(\mathbb{C})$	set $\mathfrak{F}(\mathbb{C}) = \{\mathbb{C}_t; t = 0, 1, \ldots\}$ of CEG models associated with a DCEG $\mathbb{C}$ (Chapter 7)
$\mathscr{F}$	path-cylinder σ-algebra of a DCEG $\mathbb{C}$ (Chapter 7)
$\mathscr{F}_t$	path σ-algebra of a CEG $\mathbb{C}_t$ (Chapter 7)
$\mathscr{F}(\mathbb{C})$	path-cylinder σ-algebra of a DCEG $\mathbb{C}$ (Chapter 7)
$\mathscr{F}(\mathbb{C}_t)$	path-cylinder σ-algebra of a CEG $\mathbb{C}_t$ (Chapter 7)
$\mathcal{F}o_t$	forest $\mathcal{F}o_t = \{\mathcal{T}_t(s_i); s_i \in l(\mathcal{T}_{t-1})\}$ that represents a process at time-slice t (Chapter 6)
$\mathbb{G}_0$	bipartite subgraph of an NT-DCEG that connects the subgraphs $\mathbb{D}_I$ and $\mathbb{D}_H$ (Chapter 7)
$\mathbb{G}_1$	bipartite graph that is obtained from $\mathbb{G}_0$ by a label transformation of its vertices. They provide the link between the time-slices $N-2$ and $N-1$ (Chapter 7)
$\mathbb{G}_{2(t)}$	graph that represents the dependence structure between time-slices t and $t+1, t = N-1, N, \ldots$ (Chapter 7)
$\mathcal{H}$	hyper-stage structure (Chapter 4)
$\mathcal{H}_h$	hyper-stage (Chapter 4)
I	permutation $\{1, 2, \ldots, N\} \overset{I}{\longmapsto} (i_1, i_2, \ldots, i_N)$ (Chapter 4)
$\mathcal{I}$	information (Chapter 3)
$\mathcal{I}(\lambda)$	set of indexes corresponding to a vector $\boldsymbol{\tau}(\lambda)$ (Chapter 6)
J	$J = \lvert\Psi(U)\rvert$ (Chapter 5)
J_k	$J_k = \lvert\Psi_k(U)\rvert$ (Chapter 5)
K	normalisation constant of a prior probability distribution (Chapter 5)
K^*	normalisation constant of a posterior probability distribution (Chapter 5)
$K^*(\boldsymbol{Z}^{(n)})$	normalisation constant of a posterior distribution determined by a sequence $\{\boldsymbol{Z}^{(n)}, n \geq 1\}$ (Chapter 5)

$\mathscr{L}$	partition $\mathscr{L} = \{\mathscr{L}_i\}$ of the levels of an event tree such that any two levels in the same set $\mathscr{L}_i$ are at different time-slices (Chapter 6)
L_i	number of outgoing edges associated with each situation in a stage u_i (Chapter 3)
$\mathcal{L}_\tau$	τ-norm space (Chapter 5)
$\boldsymbol{M}$	transition matrix of a Markov chain (Chapter 7)
$M+1$	number of stages in a CEG/DCEG model (Chapter 3)
$M(\boldsymbol{\mathcal{C}})$	partition of the model space $\boldsymbol{\mathcal{C}}$ such that any two CEGs in the same set M_i are Markov equivalent whilst any two CEGs chosen from different sets M_a and M_b, $a \neq b$, are not Markov equivalent (Chapter 4)
M_n	number of situations associated with the n^{th} variable in an event tree $\mathcal{T}$ (Chapter 4)
$M_{n(I)}$	number of situations associated with the n^{th} variable X_{i_n} in an $\boldsymbol{\mathcal{Z}}$-compatible event tree $\mathcal{T}(\boldsymbol{X}(I))$ (Chapter 4)
$\boldsymbol{M}_t$	transition matrix associated with the positions in $\mathbb{D}_I$ from time-slice $t, t \leq N-2$, to time-slice $N-1$ (Chapter 7)
$\boldsymbol{\mathcal{N}}(\mathbb{C})$	local neighbourhood associated with CEG $\mathbb{C}$ (Chapter 4)
$\boldsymbol{\mathcal{N}}_2(\mathbb{C})$	local neighbourhood constituted by all 1-nested CEGs in $\mathbb{C}$ (Chapter 4)
$\mathscr{P}$	probabilistic measure (Chapter 3)
$Q(\mathbb{C})$	log posterior probability of a CEG $\mathbb{C}$ (Chapter 4)
$Q(\mathcal{U}_t^{cut})$	separator random variable associated with a cut $\mathcal{U}_t^{cut}$ (Chapter 7)
$Q(\mathcal{W}_t^{cut(s)})$	separator random variable associated with a fine cut $\mathcal{W}_t^{cut(s)}$ (Chapter 7)
$Q_{U_h}(\mathbb{C})$	the score corresponding to all stages of $\mathbb{C}$ associated with the hyper-stage $\mathcal{H}_h$ (Chapter 4)
$R+1$	number of situations in an event tree (Chapter 3)

$\mathcal{S}(t+1)$	set of all situations in times-slice $t+1$ whose parent are in time-slice t (Chapter 6)
$\mathcal{T}$	event tree (Chapter 3)
$\mathcal{T}_t$	$\mathcal{T}_t \equiv \mathcal{T}_t(s_0)$ (Chapter 6)
$\mathcal{T}_{-1}$	event tree associated with a set of time-invariant covariates (Chapter 6)
$\boldsymbol{\mathcal{T}_{\mathcal{Z}}}$	set of all possible $\boldsymbol{\mathcal{Z}}-$compatible event trees (Chapter 4)
$\mathcal{T}(s_i)$	whole (finite or infinite) event tree that unfolds from a situation s_i (Chapter 6)
$\mathcal{T}_t(s_i)$	finite tree that unfolds from a situation s_i and stops at the end of interval t (Chapter 6)
$\mathcal{T}_\infty(s_i)$	whole infinite event tree that unfolds from a situation s_i (Chapter 6)
$\mathcal{T}(\boldsymbol{Z}(I))$	event tree spanned by a random vector $\boldsymbol{Z}(I)$ (Chapter 4)
U	stage structure of a CEG/DCEG model (Chapter 3)
$\mathcal{U}^{cut}$	cut at time-slice t, $t = N-1, N, \ldots$ (Chapter 7)
$\mathcal{U}_t^{cut}$	cut at time-slice t, $t = -1, 0, \ldots$ (Chapter 7)
$V(\Delta)$	point vertex set of a tree object (Chapter 6)
W	position structure of a CEG/DCEG model (Chapter 3)
$\mathcal{W}_{\mathcal{E}}$	set of positions at the beginning of time t that corresponds to $\mathcal{E}$
$\mathcal{W}_{Head}$	set of positions of an NT-DCEG for which for every position $w \in \mathcal{W}_{Head}$ there exists an edge $(w^*, w) \in E_{\textcircled{\dagger}}, w^* \in \mathbb{C}$ (Chapter 7)
$\mathcal{W}_I$	$\mathcal{W}_I = \mathcal{W}_I^{N-2}$ (Chapter 7)
$\mathcal{W}_I^t$	set of all positions of an NT-DCEG in time-slice t, $t = 0, \ldots, N-2$, that are parents of a position in time-slice $t+1$ (Chapter 7)
$\mathcal{W}_{Tail}$	set of positions of an NT-DCEG that are tails of cyclical temporal edges (Chapter 7)

$\mathcal{W}^{cut}$	fine cut at time-slice t, $t = N-1, N, \ldots$, in an NT-DCEG (Chapter 7)
$\mathcal{W}_t^{cut}$	fine cut at time-slice t, $t = -1, 0, \ldots$, in an NT-DCEG (Chapter 7)
$\mathcal{W}_t^{cut(s)}$	fine cut at time-slice t , $t = -1, 0, \ldots$, in an NT-DCEG, when a time window of analysis is limited to s time-slices ahead from t (Chapter 7)
$\mathbb{X}$	state space of a random variable X (Chapter 3)
$X(s)$	random variable associated with a situation s (Chapter 3)
$X(\mathcal{U}_t^{cut})$	downstream random variable associated with a cut $\mathcal{U}_t^{cut}$ (Chapter 7)
$X(\mathcal{W}_t^{cut(s)})$	downstream random variable associated with a fine cut $\mathcal{W}_t^{cut(s)}$ (Chapter 7)
$\boldsymbol{\mathcal{Z}}$	set of random variables $\{Z_1, Z_2, \ldots, Z_N\}$, $N \geq 2$ (Chapter 4)
$\boldsymbol{\mathcal{Z}}^k$	subset of $\boldsymbol{\mathcal{Z}}$ with k elements (Chapter 5)
$\boldsymbol{Z}^{(n)}$	$\boldsymbol{Z}^{(n)} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n)$ (Chapter 5)
$\boldsymbol{Z}_s$	random variable that registers the events that happen to the s^{th} unit in a process supported by an event tree $\mathcal{T}$ (Chapter 5)
$\boldsymbol{Z}(I)$	random vector obtained ordering the set of random variables $\boldsymbol{\mathcal{Z}}$ according to permutation I (Chapter 4)
$\mathbb{Z}^{(k)}(I)$	product space of the first k variables in $\boldsymbol{Z}(I)$ (Chapter 4)
$Z(\mathcal{U}_t^{cut})$	upstream random variable associated with a cut $\mathcal{U}_t^{cut}$ (Chapter 7)
$Z(\mathcal{W}_t^{cut(s)})$	upstream random variable associated with a fine cut $\mathcal{W}_t^{cut(s)}$ (Chapter 7)
$\boldsymbol{\alpha}^0$	hyper-parameter for a CEG $\mathbb{C}_0$ (Chapter 4)
$\boldsymbol{\alpha}_i$	phantom sample vector corresponding to a stage u_i (Chapter 3)
$\boldsymbol{\alpha}_i^*$	$\boldsymbol{\alpha}_i^*$=$\boldsymbol{\alpha}_i + \boldsymbol{x}_i$ (Chapter 3)
$\bar{\alpha}_i$	phantom sample size total associated with a stage u_i (Chapter 3)
α_{ij}	number of phantom units that transverse the stage u_i using its outgoing edge j (Chapter 3)

$\delta(\mathcal{T})$	rectangular vertex of a tree object (Chapter 6)
λ	a path in a CEG/DCEG or a walk in a DCEG (Chapter 3)
$\lambda(w_0, w)$	root-to-w path (Chapter 3)
$\boldsymbol{\mu}$	initial distribution of a Markov Chain (Chapter 7)
$\boldsymbol{\mu}(s_j)$	empirical mean conditional distributional corresponding to a situation s_j (Chapter 5)
$\boldsymbol{\phi_i}$	$\boldsymbol{\phi_i} = (\phi_{i1}, \ldots, \phi_{iL_i})$(Chapter 5)
ϕ_{ij}	probability of an individual arriving at a stage u_i and taking the emanating edge j of u_i (Chapter 5)
$\bar{\phi}_i$	visit rate of a stage u_i (Chapter 5)
$\phi(w)$	emphasis associated with evidence collection at position w (Chapter 3)
$\boldsymbol{\pi}_i$	probability vector corresponding to a stage u_i (Chapter 3)
π_{ij}	conditional probability of a unit in stage u_i unfolds through the emanating edge j of u_i (Chapter 3)
$\psi(s, s_i)$	child situation of s along the root-to-s_i path (Chapter 6)
$\boldsymbol{\tau}(\lambda)$	$\boldsymbol{\tau}(\lambda) = (\tau_i(\lambda))_{i \in \mathcal{I}(\lambda)}$ denote the ordered sequence of time-slices $\tau_i(\lambda), i \in \mathcal{I}(\lambda)$, associated with each event in a path λ (Chapter 6)
$\tau_i(w)$	potential corresponding to each propagated evidence arriving at position w through its outgoing edge i (Chapter 3)
$\xi(s_j)$	situational error (Chapter 5)
$\xi(s, k)$	time-ordered concatenation of events that precedes a situation s in the last k time-slices and in the time-invariant tree $\mathcal{T}_{-1}$ (Chapter 6)
$\xi(s_a, s_b)$	sequence of events that happen along a finite path between the situations s_a and s_b, where s_b is down stream of s_a (Chapter 6)
$\xi(\mathcal{T})$	total situational error (Chapter 5)
$\xi(\lambda)$	sequence of events associated with a walk λ in an NT-DCEG (Chapter 7)

$\Delta(\mathcal{T})$	tree object (Chapter 6)
$\Delta(U, U^+)$	set of stages of $\mathbb{C}$ that are merged in $\mathbb{C}^+$ (Chapter 4)
$\Gamma(\cdot)$	gamma function (Chapter 3)
$\Lambda(s_i)$	set of all paths in $\Lambda(\mathcal{T}_\infty)$ that pass through a situation s_i (Chapter 6)
$\Lambda(u_i)$	set of all paths in a CEG/DCEG that pass through at least one position in the stage u_i (Chapter 5)
$\Lambda_j(u_i)$	subset of all paths in $\Lambda(u_i)$ that pass through an edge j corresponding to the stage u_i (Chapter 5)
$\Lambda(w)$	set of all $w_0 - to - w$ paths in a CEG/DCEG (Chapter 3)
$\Lambda_s(w)$	set of all walks that unfolds from w over s time-slices in an NT-DCEG (Chapter 7)
$\Lambda(\mathcal{T})$	set of paths of an infinite tree $\mathcal{T}$ where the path λ is either a root-to-leaf or an infinite path of $\mathcal{T}$ (Chapter 6)
Φ	vertex contraction operation (Definition 42) (Chapter 7)
Φ_i	Bernoulli random variable of parameter $\bar{\phi}_i$ that represents whether or not an individual visits a stage u_i (Chapter 5)
Φ_{ij}	Bernoulli random variable of parameter Φ_{ij} representing whether or not an individual arrives at a stage u_i and takes its outgoing edge j (Chapter 5)
Π	set of primitive probabilities of a DCEG (Chapter 7)
Π_t	set of primitive probabilities associated with a CEG $\mathbb{C}_t$ (Chapter 7)
$\Psi(s_i)$	time-ordered concatenation of situations along the root-to-$pa(s_i)$ path (Chapter 6)
$\Psi(U)$	collection of pairs of stages (u_i, u_j) in U that can be merged to derive nested CEGs (Chapter 5)
$\Psi(\Delta(U, U^+))$	collection of pair of stages (u_i, u_j) in $\Delta(U, U^+)$ that are gathered in U^+ (Chapter 5)

$\Psi_k(U)$	largest set of stages in U yielded by a hyper-stage structure $\mathcal{H}$ such that the following property holds: for any pair of stages u_{r_1} and u_{r_2} in $\Psi_k(U)$, $(u_{r_1}, u_{r_2}) \in \Psi(U)$ (Chapter 5)
$\Psi_k(\Delta(U, U^+))$	largest set of stages in $\Delta(U, U^+)$ yielded by a hyper-stage structure $\mathcal{H}$ such that the following property holds: for any pair of stages u_{r_1} and u_{r_2} in $\Psi_k(\Delta(U, U^+))$, $(u_{r_1}, u_{r_2}) \in \Psi(\Delta(U, U^+))$ (Chapter 5)
$\Upsilon(\mathcal{T})$	$\Upsilon(\mathcal{T}) = \{\Upsilon_k; k = 1, \ldots, n\}$ (Chapter 7)
Υ_k	partition of the leaf vertex set of $\mathcal{T}$ (Chapter 6)
$\Xi_c(w(t), N)$	set of all sequences of events $\xi(s, N), s \in w(t)$, that happen along each walk from the root position w_0 to $w(t)$ whose events from time 0 to $t - N$ are excluded
$\Xi(\mathcal{W}_t^{cut(s)})$	set of sequences of events $\xi(\lambda)$ that can unfold from $\mathcal{W}_t^{cut}$ over s time-slices. When s is equal to zero, $\Xi(\mathcal{W}_t^{cut(s)})$ denotes the set of developments $\xi(\lambda)$ from $\mathcal{W}_t^{cut}$ during the current time-slice t (Chapter 7)
$\lvert\mathcal{A}\rvert$	total number of elements of a set $\mathcal{A}$ (Chapter 4)
$\lVert \cdot \rVert_\tau$	τ-norm (Chapter 5)
$\dagger$	superscript that indicates a true probability distribution (Chapter 5)
$\oplus$	graph union operation (Definition 41) (Chapter 7)
$\perp\!\!\!\perp$	stochastic independence (Chapter 7)
$\perp\!\!\!\perp_T$	T-stochastic independence (Chapter 7)
$\underrightarrow{\perp\!\!\!\perp}$	local independence (Chapter 7)
$\underrightarrow{\perp\!\!\!\perp}_T$	T-local independence (Chapter 7)
$\overset{\perp\!\!\!\perp}{\sim}$	contemporaneous independence (Chapter 7)
$\overset{\perp\!\!\!\perp_T}{\sim}$	T-contemporaneous independence (Chapter 7)
$\uplus_h$	merging operation between a tree and a set of trees (Chapter 6)

Bibliography

O. O. Aalen. Dynamic modelling and causality. *Scandinavian Actuarial Journal*, 1987(3-4):177–190, 1987.

O. O. Aalen, Kjetil Røysland, Jon Michael Gran, and Bruno Ledergerber. Causality, mediation and time: a dynamic viewpoint. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175(4):831–861, 2012.

M. Abramowitz and I. A. Stegun. Handbook of mathematical functions with formulas, graphs, and mathematical tables, 1972.

C. C. Aggarwal and C. K. Reddy, editors. *Data Clustering: Algorithms and Applications*. CRC Press, 2014.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki, editors, *Second International Symposium on Information Theory*, pages 267–281, Budapest, 1973. Akadémiai Kiado.

D. Altomare, G. Consonni, and L. La Rocca. Objective Bayesian search of Gaussian directed acyclic graphical models for ordered variables with non-local priors. *Biometrics*, 69(2):478–487, 2013.

Audit Commission. Preventing violent extremism: Learning and development exercise. Report to the Home Office and communities and local government, October 2008.

O. Bangsø and P.-H. Wuillemin. Top-down construction and repetitive structures representation in Bayesian networks. In J. Etheredge and B. Manaris, editors,

Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2000), pages 282–286. AAAI Press, 2000.

L. M. Barclay, J. L. Hutton, and J. Q. Smith. Refining a Bayesian network using a chain event graph. *International Journal of Approximate Reasoning*, 54(9):1300 – 1309, 2013.

L. M. Barclay, J. L. Hutton, and J. Q. Smith. Chain event graphs for informed missingness. *Bayesian Analysis*, 9(1):53–76, 2014.

L. M. Barclay, R. A. Collazo, J. Q. Smith, P. Thwaites, and A. Nicholson. The dynamic chain event graph. *Electronic Journal of Statistics*, 9(2):2130–2169, 2015.

R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, first edition, 1957.

J. O. Berger and L. R. Pericchi. The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996a.

J. O. Berger and L. R. Pericchi. The intrinsic Bayes factor for linear models. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Valencia International Meeting on Bayesian Statistics*, pages 23–42, Oxford, 1996b. Clarendon Press.

J. O. Berger and L. R. Pericchi. Accurate and stable Bayesian model selection: The median intrinsic Bayes factor. *Sankhy: The Indian Journal of Statistics, Series B (1960-2002)*, 60(1):1–18, 1998.

J. O. Berger and L. R. Pericchi. Objective Bayesian methods for model selection: Introduction and comparison. In P. Lahiri, editor, *Model selection*, volume Volume 38 of *Lecture Notes–Monograph Series*, pages 135–207. Institute of Mathematical Statistics, Beachwood, OH, 2001.

J. M. Bernardo and A. F. M. Smith. *Bayesian theory*. Wiley series in probability and mathematical statistics. John Wiley, Chichester, 2004.

P. Billingsley. *Convergence of probability measures*. Wiley series in Probability and Statistics. Wiley, New York ; Chichester, 2nd edition, 1999. ISBN 9780471197459.

S. Boettcher and C. Dethlefsen. deal: A package for learning Bayesian networks. *Journal of Statistical Software*, 8(1), 2003.

G. Booch. *Object-Oriented Analysis and Design with Applications*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2007.

C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In Eric Horvitz and Finn Jensen, editors, *12th Conference on Uncertainty in Artificial Intelligence (UAI 96)*, Uncertainty in Artificial Intelligence, pages 115–123, San Francisco, 1996. Morgan Kaufmann Publishers Inc.

M. Bozga and O. Maler. On the representation of probabilities over structured domains. In Nicolas Halbwachs and Doron Peled, editors, *Computer Aided Verification*, volume 1633 of *Lecture Notes in Computer Science*, pages 261–273. Springer Berlin Heidelberg, 1999.

R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, Aug 1986.

CHDS. Christchurch health and development study, 2014. URL `http://www.otago.ac.nz/christchurch/research/healthdevelopment/`. Online; accessed 27-May-2014.

H. Chipman, E. I. George, and R. E. McCulloch. The practical implementation of Bayesian model selection. In P. Lahiri, editor, *Model selection*, volume 38 of *Lecture Notes–Monograph Series*, pages 65–116. Institute of Mathematical Statistics, Beachwood, OH, 2001.

K. Christmann. Preventing religious radicalisation and violent extremism: A systematic review of the research evidence. Research report, Youth Justice Board for England and Wales, January 2012.

R. A. Collazo and J. Q. Smith. A new family of non-local priors for chain event graph model selection. *Bayesian Analysis*, 11(4):1165–1201, 12 2016. doi: 10.1214/15-BA981. URL http://dx.doi.org/10.1214/15-BA981.

R. A. Collazo and P. G. Taranti. A R package for chain event graph models. Unpublished Manuscript, 2016.

R. A. Collazo, A. Insch, G. Mcleod, J. Henry, J. van der Klei, J. Horwood, and J. Q. Smith. Using chain event graphs to understand a booking process of tourist trains in New Zealand. Unpublished Article, 2016.

G. Consonni and L. La Rocca. On moment priors for Bayesian model choice with applications to directed acyclic graphs. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 9 - Proceedings of the ninth Valencia international meeting*, pages 63–78. Oxford University Press, 2011.

G. Consonni, J. J. Forster, and L. La Rocca. The whetstone and the alum block: Balanced objective Bayesian comparison of nested models for discrete data. *Statistical Science*, 28(3):398–423, 2013.

G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2):393 – 405, 1990.

R. G. Cowell and A. P. Dawid. Fast retraction of evidence in a probabilistic expert system. *Statistics and Computing*, 2(1):37–40, 1992.

R. G. Cowell and J. Q. Smith. Causal discovery through map selection of stratified chain even graphs. Technical Report 13-14, CRiSM, 2013.

R. G. Cowell and J. Q. Smith. Causal discovery through MAP selection of stratified chain event graphs. *Electronic Journal of Statistics*, 8(1):965–997, 2014.

R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic networks and expert systems*. Statistics for engineering and information science. Springer, New York ; London, 2007.

J. Cussens. Bayesian network learning by compiling to weighted MAX-SAT. In *UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, Helsinki, Finland, July 9-12, 2008*, pages 105–112, 2008.

I. M. Cuthbertson. Prisons and the education of terrorists. *World Policy Journal*, 21(3):pp. 15–22, 2004.

J. J. Dabrowski and J. P. de Villiers. Maritime piracy situation modelling with dynamic Bayesian networks. *Information Fusion*, 23:116 – 130, 2015.

A. DasGupta. *Asymptotic Theory of Statistics and Probability*. Springer Texts in Statistics. Springer, 2008.

A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–31, 1979.

A. P. Dawid. The trouble with Bayes factors. Technical report, University College London, 1999.

A. P. Dawid. Posterior model probabilities. In Prasanta S. Bandyopadhyay and Malcolm R. Forster, editors, *Philosophy of Statistics*, volume 7, pages 607–630. North-Holland, Amsterdam, 2011.

F. de Santis and F. Spezzaferri. Methods for default and robust Bayesian model comparison: the fractional Bayes factor approach. *International Statistical Review*, 67(3):267–286, 1999.

T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.

C. Demetriou, S. Malthaner, and L. Bosi. *Dynamics of Political Violence: A Process-Oriented Perspective on Radicalization and the Escalation of Political*

Conflict. The Mobilization Series on Social Movements, Protest, and Culture. Ashgate Publishing Company, 2014.

V. Didelez. Graphical models for marked point processes based on local independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):245–264, 2008.

R. Diestel. *Graph Theory*. Electronic library of mathematics. Springer, 2006.

J. M. Dufour and E. Renault. Short run and long run causality in time series: Theory. *Econometrica*, 66(5):1099–1125, 1998.

B. Efron and A. Gous. Scales of evidence for model selection: Fisher versus Jeffreys. In P. Lahiri, editor, *Model selection*, volume 38 of *Lecture Notes–Monograph Series*, pages 208–246. Institute of Mathematical Statistics, Beachwood, OH, 2001.

M. Eichler. Granger causality and path diagrams for multivariate time series. *Journal of Econometrics*, 137(2):334 – 353, 2007.

M. Eichler and V. Didelez. On Granger causality and the effect of interventions in time series. *Lifetime Data Analysis*, 16(1):3–32, 2010.

W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, New York, NY, second edition, 1971a.

W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. Wiley, New York, NY, second edition, 1971b.

D. M. Fergusson, L. J. Horwood, A.L. Beautrais, and F. T. Shannon. Health care utilisation in a New Zealand birth cohort. *Community Health Studies*, 5(1): 53–60, 1981.

D. M. Fergusson, M. E. Dimond, L. J. Horwood, and F. T. Shannon. The utilisation of preschool health and education services. *Social Science & Medicine*, 19(11): 1173 – 1180, 1984.

D. M. Fergusson, L. J. Horwood, and F. T. Shannon. Social and family factors in childhood hospital admission. *Journal of Epidemiology and Community Health*, 40(1):50–58, 1986.

G. Freeman and J. Q. Smith. Bayesian map model selection of chain event graphs. *Journal of Multivariate Analysis*, 102(7):1152–1165, 2011a.

G. Freeman and J. Q. Smith. Dynamic staged trees for discrete multivariate time series: Forecasting, model selection and causal analysis. *Bayesian Analysis*, 6 (2):279–305, 2011b.

S. French and D. R. Insua. *Statistical decision theory: Kendall's Library of Statistics 9*. Wiley, 2010.

N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In M.I I. Jordan, editor, *Learning in Graphical Models*, pages 421–459. Springer Netherlands, Dordrecht, 1998.

D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, 82(1):45 – 74, 1996.

D. Geiger and D. Heckerman. A characterization of the Dirichlet distribution through global and local parameter independence. *The Annals of Statistics*, 25 (3):1344–1369, 1997.

D. Geiger and J. Pearl. On the logic of causal models. In R. D. Shachter, T. S. Levitt, L. N. Kanal, and J. F. Lemmer, editors, *Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '88, pages 3–14, Amsterdam, The Netherlands, 1990. North-Holland Publishing Co.

J. Geweke. Chapter 19 inference and causality in economic time series models. volume 2 of *Handbook of Econometrics*, pages 1101 – 1144. Elsevier, 1984.

P. Gill. A multi-dimensional approach to suicide bombing. *International Journal of Conflict and Violence*, 1(2):142–159, 2007.

A. Gottard. On the inclusion of bivariate marked point processes in graphical models. *Metrika*, 66(3):269–287, 2007.

C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3), 1969.

C. W. J. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2:329 – 352, 1980.

C. W. J. Granger. Some recent development in a concept of causality. *Journal of Econometrics*, 39(12):199 – 211, 1988.

E.-P. Guittet, F. Ragazzi, L. Bonelli, and D. Bigo. Preventing and countering youth radicalisation in the EU. Research report, European Parliament, April 2012.

G. Hannah, L. Clutterbuck, and J. Rubin. Radicalization or rehabilitation. Understanding the challenge of extremist and radicalized prisoners. Technical Report TR 571, RAND Corporation, 2008.

P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1):191–215, 1997.

N. A. Heard, C. C. Holmes, and D. A. Stephens. A quantitative study of gene regulation involved in the immune response of anopheline mosquitoes: An application of Bayesian hierarchical clustering of curves. *Journal of the American Statistical Association*, 101(473):18–29, 2006.

D. Heckerman. A tutorial on learning with Bayesian networks. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 301–354. MIT Press, Cambridge, MA, USA, 1999.

D. Heckerman. A tutorial on learning with Bayesian Networks. *Innovations in Bayesian Networks*, pages 33–82, 2008.

D. Heckerman and D. Geiger. Learning Bayesian networks: A unification for discrete and Gaussian domains. In *UAI '95: Proceedings of the Eleventh Annual*

Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, Canada, August 18-20, pages 274–284, 1995.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3): 197–243, 1995.

C. Howson and P. Urbach. Probability, uncertainty and the practice of Statistics. In G. Wright and P. Ayton, editors, *Subjective Probability*, pages 39–51. John Wiley & Sons, Oxford, England, 1994.

C. Hsiao. Autoregressive modeling and causal ordering of economic variables. *Journal of Economic Dynamics and Control*, 4:243 – 259, 1982.

M. Jaeger. Probabilistic decision graphs - combining verification and AI techniques for probabilistic inference. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 12:19–42, 2004.

M. Jaeger, J. D. Nielsen, and T. Silander. Learning probabilistic decision graphs. *International Journal of Approximate Reasoning*, 42(1-2):84–100, 2006.

A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall Advanced Reference Series. Prentice Hall PTR, 1988.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

V. E. Johnson and D. Rossell. On the use of non-local prior densities in Bayesian hypothesis tests. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 72:143–170, 2010.

V. E. Johnson and D. Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(500):1656–1656, 2012.

J. Jordan and N. Horsburgh. Spain and Islamist terrorism: Analysis of the threat and response 1995-2005. *Mediterranean Politics*, 11(2):209–229, 2006.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

N. Khakzad. Application of dynamic Bayesian network to risk analysis of domino effects in chemical infrastructures. *Reliability Engineering & System Safety*, 138: 263 – 272, 2015. ISSN 0951-8320.

U. Kjærulff. A computational scheme for reasoning in dynamic probabilistic networks. In *Proceedings of the Eighth International Conference on Uncertainty in Artificial Intelligence*, UAI'92, pages 121–129, 1992.

I. Klugkist and H. Hoijtink. The Bayes factor for inequality and about equality constrained models. *Computational Statistics and Data Analysis*, 51(12):6367–6379, 2007.

M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts, USA, 2009.

D. Koller and A. Pfeffer. Object-oriented Bayesian networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, UAI'97, pages 302–313, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

K. B. Korb and A. E. Nicholson. *Bayesian artificial intelligence*. Chapman and Hall/CRC computer science and data analysis series. CRC Press, Boca Raton, FL, second edition, 2011.

P. J. Krause and D. A. Clark. Uncertainty and subjective probability in AI systems. In G. Wright and P. Ayton, editors, *Subjective Probability*, pages 501–527. John Wiley & Sons, Oxford, England, 1994.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H.-G. Leimer. Independence properties of directed Markov fields. *Networks*, 20(5):491–505, 1990.

P. Li, P. Gong, H. Li, E. J. Perkins, N. Wang, and C. Zhang. Gene regulatory network inference and validation using relative change ratio analysis and time-delayed dynamic Bayesian network. *EURASIP Journal on Bioinformatics and Systems Biology*, 2014(1):12, 2014.

D. Lindley. Foundations. In G. Wright and P. Ayton, editors, *Subjective Probability*, pages 3–15. John Wiley & Sons, Oxford, England, 1994.

S. Liverani, J. Cussens, and J. Q. Smith. Searching a multivariate partition space using MAX-SAT. In F. Masulli, L. E. Peterson, and R. Tagliaferri, editors, *Computational Intelligence Methods for Bioinformatics and Biostatistics*, volume 6160 of *Lecture Notes in Computer Science*, pages 240–253. Springer Berlin Heidelberg, 2010.

H. Ltkepohl. Testing for causation between two variables in higher-dimensional var models. In H. Schneeweiß and K. F. Zimmermann, editors, *Studies in Applied Econometrics*, Contributions to Economics, pages 75–91. Physica-Verlag HD, 1993.

D. J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge, UK ; New York, 2003.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

S. Marini, E. Trifoglio, N. Barbarini, F. Sambo, B. Di Camillo, A. Malovini, M. Manfrini, C. Cobelli, and R. Bellazzi. A dynamic Bayesian network model for long-term simulation of clinical complications in type 1 diabetes. *Journal of Biomedical Informatics*, 57:369 – 376, 2015.

D. McAllester, M. Collins, and F. Pereira. Case-factor diagrams for structured probabilistic modeling. *Journal of Computer and System Sciences*, 74(1):84–96, 2008.

C. McCauley and S. Moskalenko. Mechanisms of political radicalization: Pathways toward terrorism. *Terrorism and Political Violence*, 20(3):415–433, 2008.

Ministry of Justice. Annual tables - offender management caseload statistics 2012 tables, 2013. URL `https://www.gov.uk/government/statistics/offender-management-statistics-quarterly--2`. Online; accessed 03-Nov-2014.

F. M. Moghaddam. The staircase to terrorism: A psychological exploration. *American Psychologist*, 60:161–169, 2005.

E. Moreno. Bayes factors for intrinsic and fractional priors in nested models. Bayesian robustness. L_1-*Statistical Procedures and Related Topics*, 31:257–270, 1997.

E. Moreno, F. Bertolino, and W. Racugno. An intrinsic limiting procedure for model selection and hypotheses testing. *Journal of the American Statistical Association*, 93(444):1451–1460, 1998.

R. E. Neapolitan. *Learning Bayesian networks*. Prentice Hall, Harlow, 2004.

M. Neil, N. Fenton, and L. Nielson. Building large-scale Bayesian networks. *The Knowledge Engineering Review*, 15(3):257–284, September 2000.

P. O. B. Netto. *Grafos: Teorias, modelos, algoritmos*. Blucher, 2006.

P. E. Neumann. Prisons and terrorism: Radicalisation and de-radicalisation in 15 countries. Technical report, International Centre for the Study of Radicalisation and Political Violence, London, July 2010.

A. E. Nicholson. *Monitoring Discrete Environments Using Dynamic Belief Networks*. PhD thesis, Department of Engineering Sciences, Oxford, 1992.

U. Nodelman, C. R. Shelton, and D. Koller. Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 378–387, 2002.

U. Nodelman, C. R. Shelton, and D. Koller. Learning continuous time Bayesian networks. In *Proceedings of the Nineteenth International Conference on Uncertainty in Artificial Intelligence*, pages 451–458, 2003.

A. O'Hagan. Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society Series B-Methodological*, 57(1):99–138, 1995.

A. O'Hagan. Properties of intrinsic and fractional Bayes factors. *Test*, 6(1): 101–118, 1997.

A. O'Hagan and J. Forster. *Bayesian inference*. Kendall's advanced theory of statistics. Arnold, London, 2nd edition, 2004.

S. Ott and S. Miyano. Finding optimal gene networks using biological constraints. *Genome informatics. International Conference on Genome Informatics*, 14:124–33, 2003.

J. Pearl. A constraint-propagation approach to probabilistic reasoning. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 357–370, Amsterdan, The Netherlands, 1986. Elsevier.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

J. Pearl. *Causality : models, reasoning, and inference*. Cambridge University Press, Cambridge, 2009.

J. Pearl and A. Paz. Graphois: A graph-based logic for reasoning about relevance relations. In B. Du Boulay, D. Hogg, and L. Steels, editors, *Advances in Artificial Intelligence - II*, pages 357–363, Amsterdan, The Netherlands, 1987. North-Holland Publishing Co.

Y. Peres. Probability on trees: An introductory climb. In P. Bernard, editor, *Lectures on Probability Theory and Statistics*, volume 1717 of *Lecture Notes in Mathematics*, pages 193–280. Springer Berlin Heidelberg, 1999.

J. M. Perez and J. O. Berger. Expected-posterior prior distributions for model selection. *Biometrika*, 89(3):491–511, 2002.

L. R. Pericchi. Model selection and hypothesis testing based on objective probabilities and Bayes factors. In D. K. Dey and C. R. Rao, editors, *Bayesian Thinking, Modeling and Computation*, volume 25 of *Handbook of Statistics*, pages 115–149. Elsevier, 2005.

L. D. Phillips. A theory of requisite decision models. *Acta Psychologica*, 56(1):29 – 48, 1984.

L. D. Phillips. Decision conferencing. In W. Edwards, R. F. Miles, and D. von Winterfeldt, editors, *Advances in Decision Analysis: From Foundations to Applications*, pages 375–399. Cambridge University Press, 2007.

D. Poole and N. L. W. Zhang. Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research*, 18:263–313, 2003.

O. Pourret, P. Naïm, and B. Marcot. *Bayesian Networks: A Practical Guide to Applications*. Statistics in Practice. Wiley, 2008.

T. Precht. *Home Grown Terrorism and Islamist Radicalisation in Europe: From Conversion to Terrorism : an Assessment of the Factors Influencing Violent Islamist Extremism and Suggestions for Counter Radicalisation Measures*. Ministry of Justice, 2007.

C. R. Rao. A review of canonical coordinates and an alternative to correspondence analysis using Hellinger distance. *Questiio*, 19(1-3):23–63, 1995.

C. R. Rao and Y. Wu. On model selection. In P. Lahiri, editor, *Model selection*, volume 38 of *Lecture Notes–Monograph Series*, pages 1–57. Institute of Mathematical Statistics, Beachwood, OH, 2001.

D. Rossell and D. Telesca. Non-local priors for high-dimensional estimation. *Journal of the American Statistical Association*, pages 1–33, 2015. doi: 10.1080/01621459.2015.1130634. URL http://dx.doi.org/10.1080/01621459.2015.1130634.

J. Rousseau. Approximating interval hypothesis : p-values and Bayes factors. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 8 - Proceedings of the eighth Valencia international meeting*, pages 417–452, Oxford, 2007. Oxford University Press.

R. Rowe. From jail to jihad? The threat of prison radicalisation, 12 May 2014. URL http://www.bbc.co.uk/news/uk-27357208. Online; published 12-May-2014, accessed 19-Jan-2015.

F. Rubio, M J. Flores, J. M. Gómez, and A. Nicholson. Dynamic Bayesian networks for semantic localization in robotics. In *XV Workshop of physical agents: book of proceedings, WAF 2014, June 12th and 13th, 2014 León, Spain*, pages 144–155, 2014.

M. J. Schervish. *Theory of Statistics*. Springer Series in Statistics. Springer New York, 1996.

A. P. Schmid. Radicalisation, de-radicalisation, counter-radicalisation: A conceptual discussion and literature review. Research paper, The International Centre for Counter-Terrorism - The Hague 4, 2013.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 03 1978.

T. Schweder. Composable Markov processes. *Journal of Applied Probability*, 7(2): 400–410, 1970.

R. Segala. *Modeling and Verification of Randomized Distributed Real-time Systems*. PhD thesis, Cambridge, MA, USA, 1995.

G. Shafer. *The Art of Causal Conjecture*. Artificial Management. MIT Press, 1996.

S. E. Shimony. Finding maps for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399 – 410, 1994.

T. Silander and T.-Y. Leong. A dynamic programming algorithm for learning chain event graphs. In J. Fürnkranz, E. Hüllermeier, and T. Higuchi, editors, *Discovery Science*, volume 8140 of *Lecture Notes in Computer Science*, pages 201–216. Springer Berlin Heidelberg, 2013.

T. Silander and P. Myllymaki. A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the Twenty-Second Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 445–452, Arlington, Virginia, 2006. AUAI Press.

T. Silander, P. Kontkanen, and P. Myllymaki. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. 2007.

M. D. Silber and A. Bhatt. Radicalization in the West: The home-grown threat. Technical Report, New York City Police Department, 2007.

A. Silke. *The psychology of counter-terrorism*. Cass series on political violence. Routledge, London, England, 2011.

A. Singh and A. Moore. Finding optimal Bayesian networks by dynamic programming. Technical report, Carnegie Mellon University, June 2005.

J. Q. Smith. *Bayesian decision analysis: principles and practice*. Cambridge University Press, Cambridge, New York, 2010.

J. Q. Smith and P. E. Anderson. Conditional independence and chain event graphs. *Artificial Intelligence*, 172(1):42–68, 2008.

D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, 1990.

M. Z. Spivey. A generalized recurrence for Bell numbers. *Journal of Integer Sequences*, 11(2), 2008.

W. Spohn. Stochastic independence, causal independence, and shieldability. *Journal of Philosophical Logic*, 9(1):73–99, 1980.

H. Steck. Learning the Bayesian network structure: Dirichlet prior vs data. In *UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, Helsinki, Finland, July 9-12, 2008*, pages 511–518, 2008.

H. Steck and T. S. Jaakkola. On the Dirichlet prior and Bayesian regularization. In *In Advances in Neural Information Processing Systems 15*, pages 697–704. MIT Press, 2002.

M. Stoelinga. An introduction to probabilistic automata. *Bulletin of the EATCS*, 78:176–198, 2002.

J. Sun and J. Sun. A dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data. *Transportation Research Part C: Emerging Technologies*, 54:176 – 186, 2015.

P. Thwaites. Causal identifiability via chain event graphs. *Artificial Intelligence*, 195:291–315, 2013.

P. Thwaites, J. Q. Smith, and E. Riccomagno. Causal analysis with chain event graphs. *Artificial Intelligence*, 174(12-13):889–909, 2010.

P. A. Thwaites and J. Q. Smith. Evaluating causal effects using Chain Event Graphs. In *Proceedings of Probabilistic Graphical Models*, pages 293–300, Prague, Czech Republic, 2006a.

P. A. Thwaites and J. Q. Smith. Non-symmetric models, Chain Event Graphs and propagation. In *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 2339–2347, Paris, France, 2006b.

P. A. Thwaites and J. Q. Smith. Separation theorems for Chain Event Graphs. *CRiSM Research Report 11-09*, 2011.

P. A. Thwaites, J. Q. Smith, and R. G. Cowell. Propagation using chain event graphs. In *Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 546–553, Corvallis, Oregon, 2008. AUAI Press.

P. A. Thwaites, G. Freeman, and J. Q. Smith. *Chain Event Graph MAP model selection*. Proceedings of the International Conference on Knowledge Engineering and Ontology Development. Funchal, Portugal, 2009.

I. Verdinelli and L. Wasserman. Bayes factors, nuisance parameters and imprecise tests. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Valencia International Meeting on Bayesian Statistics*, pages 765–771, Oxford, 1996. Clarendon Press.

T. Verma and J. Pearl. Causal networks: Semantics and expressiveness. In R. D. Shachter, T. S. Levitt, L. N. Kanal, and J. F. Lemmer, editors, *Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '88, pages 69–76, Amsterdam, The Netherlands, 1990. North-Holland Publishing Co.

M. West and J. Harrison. *Bayesian forecasting and dynamic models*. Springer series in Statistics. Springer, New York, 2nd edition, 1999.

Q. Wiktorowicz. Joining the cause: Al-Muhajiroun and radical Islam. Research paper, Department of International Studies, Rhodes College, 2004.

L. A. Wolsey. *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1998.

L. A. Wolsey and G. L. Nemhauser. *Integer and Combinatorial Optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2014.

G. Wright and P. Ayton. *Subjective Probability*. John Wiley & Sons, Oxford, England, 1994.

R. Xu and D. C. Wunsch. *Clustering*. IEEE series on computational intelligence. Wiley, 2009.